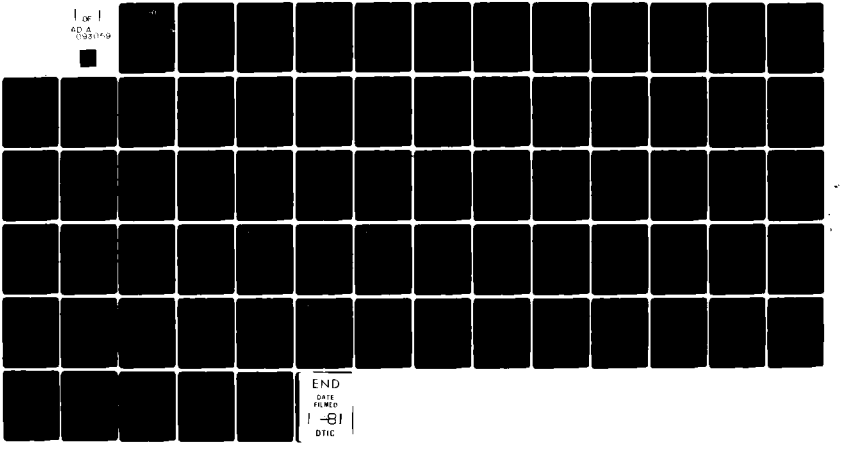
AD-A093 059 BELL AEROSPACE TEXTRON BUFFALO N Y F/G 20/5
NUMERICAL MODEL DEVELOPMENT FOR LASER CAVITY FLOWFIELDS.(U)
MAY 80 J T SCHIMKE F49620-77-C-0076
UNCLASSIFIED AFOSR-TR-80-1206 NL



FINAL

NUMERICAL MODEL DEVELOPMENT FOR LASER CAVITY FLOWFIELDS

FOR PERIOD COVERING MAY 31, 1979 TO MAY 31, 1980

J. T. SCHIMKE

FOR

DEPARTMENT OF THE AIR FORCE
AIR FORCE OFFICE OF SCIENTIFIC RESEARCH (AFSC)
BOLLING AIR FORCE BASE, DC 20332

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER (18) AFOSR-TR-80-1206	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) Numerical Model Development for Laser Cavity Flowfields		5. TYPE OF REPORT & PERIOD COVERED (9) Final Yr. 1 31 May 79 - 31 May 80
7. AUTHOR(s) (10) J. T. Schimke		8. CONTRACT OR GRANT NUMBER(s) (15) F49620-77-C-0076
9. PERFORMING ORGANIZATION NAME AND ADDRESS Bell Aerospace Textron, Div. of Textron Inc. P. O. Box 1 Buffalo, NY 14240		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (10) (11) 61102F 2304/A3
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Office of Scientific Research Bolling AFB Washington, DC 20332		12. REPORT DATE (11) May 1980
		13. NUMBER OF PAGES 66 (12) 51
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Finite Differences Adaptive Mesh Burger's Equation Implicit Integration

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A recently developed subroutine package for the integration of systems of differential-algebraic equations by implicit methods is used to compute solutions to Burger's equation. Burger's equation is discretized in space by finite difference methods to produce the differential-algebraic system. The finite differencing scheme provides for the continous automatic variation of the finite difference mesh during the course of the integration. A variety of numerical examples is presented and comparison is made with previously computed fixed mesh solutions.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

NUMERICAL MODEL DEVELOPMENT FOR LASER CAVITY FLOWFIELDS

FOR PERIOD COVERING MAY 31, 1979 TO MAY 31, 1980

J. T. SCHIMKE

FOR

DEPARTMENT OF THE AIR FORCE
AIR FORCE OFFICE OF SCIENTIFIC RESEARCH (AFSC)
BOLLING AIR FORCE BASE, DC 20332

SUMMARY

This report describes the results of our efforts on AFOSR Contract F49620-77-C-0076, Numerical Model Development for Laser Cavity Flowfields. The research was aimed toward developing a technique for automatically varying (in time) the one-dimensional spatial finite difference mesh over which certain time dependent partial differential equations were discretized. In particular, the variable mesh points should adapt themselves to the solution that is being marched out in time, i.e., they should be closely spaced in regions where there are large spatial gradients and widely spaced where the solution is smooth. Further, this automatic mesh variation method is to be designed to be used in conjunction with the GEARIB implicit integration package, which was used successfully for a variety of fixed mesh discretizations during the first two years of this contract. (GEARIB is used march out in time the solution of the system of ordinary differential equa ons resulting from spatially discretizing the governing PDE(s).)

The partial differential equation that was used as a test bed for this automatic mesh variation was Burger's equation. Numerical techniques (using GEARIB on a fixed mesh) for solving this PDE were developed during the first year of this contract. Although Burger's equation is simple in appearance, its solutions have many of the characteristics (nonlinearity, wave-like propagation, diffusive effects, and steep gradients) that are found in solutions of laser flow equations, which are the problems that the methods developed herein are ultimately intended for.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (CONT'D)

I. ADAPTIVE MESH FOR BURGER'S EQUATION

A. Introduction

In the report on the results of the first year of this study, Reference [1], a detailed discussion was given in Section II regarding the choice of an integration method to march out the solution in the time-like coordinate. It was concluded that an implicit integration method would be required to efficiently handle differential systems containing the strong stiffness effects due to chemistry and radiation present in the numerical solution of laser flows. It was also concluded that the GEARIB package [2], which embodies a family of implicit methods and which is also designed to treat implicit differential-algebraic systems was well suited to our applications.

During the second year of this study, numerical difficulties arose in the treatment of the laser flow equations because the onset of lasing was accompanied by extreme spatial gradients in some of the dependent variables being integrated. The location of these gradients is unknown a priori, and may in general vary in time. In order to obtain solutions, a trial and error mesh selection process was used; pick a mesh, solve the problem, refine the mesh based on the current solution, solve the problem again, etc. The only way to avert such an iterative procedure would be to use a uniformly fine spatial mesh. Such a mesh adequate to accommodate the gradients of realistic laser flows would be computationally prohibitive. Thus, it became apparent that a successful numerical treatment of such flows would require a finite difference mesh that automatically adapts to the solution.

This requirement for an adaptive spatial finite difference mesh has been cited previously in the literature. For example, Dwyer and Sanders [4] in their study of unsteady combustion phenomena suggest an approach to an adaptive finite difference mesh but do not actually use it. (All the results they present are for uniform meshes.) Chong [5] uses adaptive meshes in his study of Burger's equation, which has solutions with steep spatial gradients. Although he obtains good results, his approach does not seem practical for our applications. The main reason is that Chong adjusts his finite difference mesh at discrete times, which would mean in our case that the GEARIB integrator would have to be restarted

after each such adjustment. This is a rather severe penalty to pay especially if GEARIB had been operating with a high order method prior to the rediscretization, since restart begins with a first order method and, consequently, a rather small step size. A similar approach has been taken by Miller, Morton, and Baines [6] in a moving boundary problem governed by a generalized one-dimensional diffusion equation.

In the approach described in this report, the rediscretization is done on a continuous basis, and so there is no need to restart the integrator (as there is when the spatial mesh is altered at discrete times). A further advantage of this approach is that moving boundary problems are readily accommodated.

B. Discretization of Burger's Equation

One of the simplest nonlinear PDE's which contains both convective and diffusive effects is Burger's equation,

$$\frac{\partial T}{\partial t} = \beta \frac{\partial^2 T}{\partial x^2} - \gamma T \frac{\partial T}{\partial x} \tag{1}$$

where β and γ are constants which can be varied to emphasize the diffusive or convective contributions. Because Burger's equation has known exact solutions which are traveling waves with very steep fronts (like shock waves) it has been a popular equation to test numerical schemes on. In fact, during the first year of this study a detailed numerical treatment of this equation was undertaken (see Chapter IV of Schimke, Rushmore, and Zelazny [1] for details). Because of this experience with Burger's equation, it seems an ideal test bed for our proposed adaptive mesh scheme.

As in Reference [1], we begin by introducing the auxiliary variable, v,

$$v \equiv \frac{\partial T}{\partial x} \tag{2}$$

Then (1) can be replaced by

$$\frac{\partial T}{\partial t} = \beta \frac{\partial v}{\partial x} - \gamma v T \tag{3a}$$

$$0 = v - \frac{\partial T}{\partial x} \tag{3b}$$

The boundary conditions are assumed to be separated but otherwise quite general, i.e.,

$$g_1(T_1,v_1,t) = 0, \qquad g_N(T_N,v_N,t) = 0 \tag{4}$$

where g_1 and g_N are arbitrary functions continuous in t, and the subscripts 1 and N refer to the left and right boundaries, which may be moving in time.

Now assume a finite difference mesh with nodes at x_1, x_2, ..., x_N (which are in general functions of time), with T_i and v_i associated with x_i, and $H_i \equiv (x_{i+1} - x_i)$. Now since T_i corresponds to a mesh point which is moving, the rate of change of T_i in time (i.e., $\dot{T}_i$) is not the same as $[\partial T/\partial t]_{x_i}$ in (3a). The relation between these two quantities is:

$$\left[\frac{\partial T}{\partial t}\right]_{x_i} = \dot{T}_i - \dot{x}_i\left[\frac{\partial T}{\partial x}\right]_{x_i} = \dot{T}_i - \dot{x}_i v_i \tag{5}$$

Now the x_i are to be determined *automatically during the course of the* problem so we must establish conditions which will uniquely specify them. For x_1 and x_N we assume,

$$x_1 = f_1(t), \qquad x_N = f_N(t) \tag{6}$$

where f_1 and f_N are prescribed functions of time. f_1 and f_N could be readily generalized to be functions of the dependent variables without complicating the numerical process. Such f_1 and f_N are required in certain types of moving boundary problems, but we will not consider them here.

To complete the system of equations, relationships specifying the interior x_i must be established. This specification is the real crux of the numerical problem. We certainly expect that a suitable mesh spacing should vary inversely with the local gradients, i.e., the local values of v_i. Such a relationship can be written:

$$(x_{i+1} - x_i) = \frac{\Delta}{F(v_i,v_{i+1})} \tag{7}$$

where F is a function that can be specified in various ways and Λ is a positive function of t but is constant in x. Λ in effect provides a uniform scaling of all the $(x_{i+1} - x_i)$, i = 1,N-1 so that the mesh "fits into" the region $[x_1, x_N]$. F in (7) could have been made a function of T_i and T_{i+1} (in addition to v_i and v_{i+1}) without introducing any essential complication, but we have had no occasion to do so during the course of this study. The relation (7) involves values at two adjacent mesh points. Why not allow a more general relation involving three (or more) consecutive mesh points? This would be contrary to the philosophy which led to the introduction of the auxiliary dependent variable, v, in equation (2). Introduction of v allows a simple differencing scheme to be employed for the partial differential system (10). This differencing scheme is unaffected by mesh nonuniformity, requires no modification next to boundary points, and results in a Jacobian matrix* with a narrow bandwidth.

According to equation (7), $(x_{i+1} - x_i)$ is given explicitly as a function of Λ, v_i, and v_{i+1}. In practice, it is more convenient to specify $(x_{i+1} - x_i)$ implicitly. Instead of (7) we use

$$(x_{i+1} - x_i) = \frac{\Lambda}{F(x_i, x_{i+1}, v_i, v_{i+1})} \tag{8}$$

A specific example of (8) used in the course of this study is:

$$(x_{i+1} - x_i) = \frac{\Lambda}{\sqrt{\left(\frac{v_{i+1} - v_i}{x_{i+1} - x_i}\right)^2 + \varepsilon}} \tag{9}$$

The squared term in (9) is an approximation to $\partial^2 T/\partial x^2$. Thus, (9) requires that the mesh become finer in regions of large $\partial^2 T/\partial x^2$, a reasonable criterion. The positive constant ε is present to prevent the mesh spacing from becoming arbitrarily large in regions of small $\partial^2 T/\partial x^2$. Note that $F(x_i, x_{i+1}, v_i, v_{i+1})$ as defined in (9) is always positive, thereby guaranteeing that consecutive mesh points will always be distinct. (Λ can never vanish.)

*Implicit integration methods (as in GEARIB) require the formation and factorization of the Jacobian matrix of the differential system.

When Λ was introduced in (7), it was described simply as a scaling factor. But it also has a "geometric" interpretation. Rewriting (8) and summing over the (N-1) mesh intervals between x_1 and x_N we get:

$$\Lambda = \frac{1}{(N-1)} \sum_{i=1}^{N-1} F(x_i,x_{i+1},v_i,v_{i+1})(x_{i+1} - x_i) \approx \frac{1}{(N-1)} \int_{x_1}^{x_N} F(x_i,x_{i+1},v_i,v_{i+1})dx \qquad (10)$$

Now $F(x_i,x_{i+1},v_i,v_{i+1})$ is normally selected to be some measure of the "variability" of the solution within $[x_i,x_{i+1}]$. For example, in (9) as $\varepsilon \to 0$

$$F = \left|\frac{\partial^2 T}{\partial x^2}\right|^{\frac{1}{2}} \qquad (11)$$

Thus, Λ is a measure of the average variability over a mesh interval.

Now differencing the partial differential system (3) in an obvious way, using (4), (5), (6), and (8), and adopting the notation,

$$H_i \equiv x_{i+1} - x_i \qquad (12a)$$

$$\bar{v}_{ij} \equiv (v_i + v_j)/2 \qquad (12b)$$

$$\bar{T}_{ij} \equiv (T_i + T_j)/2 \qquad (12c)$$

we obtain the differential-algebraic system on the following page.

Notice that product terms such as vT in (3a) and $\dot{x}v$ in (5) have been approximated as products of sums in (13) (instead of sums of products). For example, on mesh interval i, we wrote the term vT as $\bar{v}_{i,i+1}\,\bar{T}_{i,i+1} \equiv (v_i + v_{i+1})(T_i + T_{i+1})/4$ instead of $(v_iT_i + v_{i+1}\,T_{i+1})/2$. Numerical experiments were performed using both formulations. The results, although not conclusive, tended to favor the product of sums formulation. Possibly the reason is that the effect of spurious high frequency oscillations (especially in v_i) tend to be negated when using products of sums.

$$
\begin{bmatrix}
0 & 0 & 0 & & & & & & & & & & & & & \\
0 & 0 & 0 & & & & & & & & & & & & & \\
1 & 0 & -\bar{v}_{12} & 1 & 0 & -\bar{v}_{12} & & & & & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & & & & & & & & & & \\
0 & 0 & 0 & 0 & 0 & 0 & & & & & & & & & & \\
 & & & 1 & 0 & -\bar{v}_{23} & 1 & 0 & -\bar{v}_{23} & & & & & & & \\
 & & & 0 & 0 & 0 & 0 & 0 & 0 & & & & & & & \\
 & & & 0 & 0 & 0 & 0 & 0 & 0 & & & & & & & \\
 & & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & & \\
 & & & & & & & & & 1 & 0 & -\bar{v}_{N-1,N} & 1 & 0 & -\bar{v}_{N-1,N} & 0 \\
 & & & & & & & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & & & & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & & & & & & & 0 & 0 & 0 & 0 \\
 & & & & & & & & & & & & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{Bmatrix}
\dot{T}_1 \\ \dot{v}_1 \\ \dot{x}_1 \\ \dot{T}_2 \\ \dot{v}_2 \\ \dot{x}_2 \\ \dot{T}_3 \\ \dot{v}_3 \\ \dot{x}_3 \\ \\ \dot{T}_{N-1} \\ \dot{v}_{N-1} \\ \dot{x}_{N-1} \\ \dot{T}_N \\ \dot{v}_N \\ \dot{x}_N \\ \dot{\Lambda}
\end{Bmatrix}
=
\begin{Bmatrix}
g_1(T_1, v_1, t) \\
x_1 - f_1(t) \\
2\beta(v_2 - v_1)/H_1 - 2\gamma\bar{v}_{12}\bar{T}_{12} \\
(v_1 + v_2) - 2(T_2 - T_1)/H_1 \\
(x_2 - x_1) - \Lambda/F(x_1, x_2, v_1, v_2) \\
2\beta(v_3 - v_2)/H_2 - 2\gamma\bar{v}_{23}\bar{T}_{23} \\
(v_2 + v_3) - 2(T_3 - T_2)/H_2 \\
(x_3 - x_2) - \Lambda/F(x_2, x_3, v_2, v_3) \\
\\ \\ \\
2\beta(v_N - v_{N-1})/H_{N-1} - 2\gamma\bar{v}_{N-1,N}\bar{T}_{N-1,N} \\
(v_{N-1} + v_N) - 2(T_N - T_{N-1})/H_{N-1} \\
(x_N - x_{N-1}) - \Lambda/F(x_{N-1}, x_N, v_{N-1}, v_N) \\
g_N(T_N, v_N, t) \\
x_N - f_N(t)
\end{Bmatrix}
\qquad (13)
$$

Although the matrix multiplying the derivative vector is singular, this presents no problems for the GEARIB integrator. (Similar singular differential-algebraic systems were solved during the first two years of this study.) One special characteristic of system (13) which was not present in the numerical treatment of Burger's equation on a fixed mesh, is that the Jacobian of the system is no longer strictly banded. The last column has many non-zero elements. Also, certain variants of system (13) having many non-zero elements in the last row of the Jacobian have been tested. Consequently, it was necessary to modify GEARIB (which is tailored to systems with strictly banded Jacobians) to account for the possibly full last row and column. The GEARIB LU decomposition routine was modified so that the upper (N-1)×(N-1) is treated as a band matrix and special formulas are used for the last row and column of the L and U factors, respectively. With this modified LU decomposition plus a few changes within some of the other routines of the GEARIB package, such systems with "almost" banded Jacobians are efficiently integrated.

C. Specification of Boundary and Initial Conditions

In Section IV.C.1 of [1] an extensive discussion is given regarding the specification of boundary and initial conditions when solving Burger's equation on a fixed mesh. It is shown that it is important to choose initial and boundary conditions so that all strictly algebraic equations are satisfied initially. On a fixed mesh, this is easy to do for the boundary conditions

$$v_1 = g_1(t), \qquad T_N = g_N(t), \text{ or} \tag{14a}$$

$$T_1 = g_1(t), \qquad v_N = g_N(t) \tag{14b}$$

Then the algebraic equations arising from the discretization of (3b), i.e.,

$$(v_i + v_{i+1}) = (T_i + T_{i+1})/H_i \tag{15}$$

are satisfied by solving (15) recursively for v_i (assuming T_i is prescribed which is the usual case). For boundary conditions other than (14), the initial satisfaction of all algebraic equations is less straightforward. The additional

complications arising out of such boundary conditions and methods for dealing with them are detailed in [1].

For variable meshes, there is the additional set of algebraic equations obtained by writing (8) at the mid points of each mesh interval, i.e., equations (5), (8), ..., (N-2) of system (13). These equations (e.g., (9)) are nonlinear. Consequently, their initial solution poses more of a problem. In view of this, all of the examples described in this report use boundary conditions (14a)* in order not to compound the problem of determining initial conditions and therefore deter us from our main goal of devising a reliable scheme for automatically adapting the finite difference mesh to the solution as it evolves in time.

Another assumption that is made in all the examples of this report is that the initial values of T are given by a function of x which is continuous and piecewise differentiable. The method for determining initial conditions for x_i and v_i described in the following is based on this assumption. (A discussion of the problem where the initial T_i are assigned values at discrete x_i is given at the end of this section.)

A modified Newton method was used to solve the system of nonlinear equations associated with the determination of the initial conditions. The GEARIB package is designed to solve differential-algebraic systems, which includes purely algebraic systems. Thus, the solution for the initial conditions can be done within the framework of GEARIB. Certain changes to the standard GEARIB scheme were found useful to increase the likelihood of convergence.

For the determination of the initial conditions, the governing system is (13) with the matrix on the LHS set to zero and with equations (3), (6), ..., (N-4) replaced by

$$0 = T_i - T(x_i) \qquad (16)$$

*(14a) was chosen over (14b) since our sample problems all have wave-like solutions which travel in the positive x-direction. Consequently, the "upstream" boundary condition tends to have a less important effect on the solution, especially when the wave impinges on the downstream boundary. Thus, we have prescribed the more usual condition (on T rather than v) at the downstream boundary.

where T(x) is the function defining the initial values of T as a continuous function of x. To distinguish the iterations for the initial conditions within GEARIB, an artificial time starting with -50 and incremented by 1 on each call to GEARIB was established. (Integration of the actual differential system began at t=0.) This allows 50 iterations for the initial conditions to converge. Convergence typically took place within 3 to 12 iterations. The problem was aborted if 50 iterations did not produce convergence.

GEARIB requires a number of user coded routines to define the system to be solved. Within each of these routines there is at least one test on t. If t is negative, branches are taken to define the system for the initial conditions; for t positive, branches are taken to define system (13). Error tests, convergence tests, and the Jacobian re-evaluation criterion in GEARIB are modified so that only one Newton Iteration (with a newly evaluated Jacobian) is performed for each call to GEARIB when solving for initial conditions. (Testing for convergence of the iterations is performed in the calling program.)

The specific steps of the initial conditions iteration procedure are as follows:

1. Set time to -50 and step size to 1.
2. Choose x_i to be equally spaced on the interval $[x_1, x_N]$.
3. Evaluate T_i from (16).
4. Solve for v_i recursively using (15).
5. Solve for Λ using

$$\Lambda = (x_N - x_1) \Big/ \sum_{i=1}^{N-1} \left\{ 1/F(x_i, x_{i+1}, v_i, v_{i+1}) \right\}$$

   where F is as in (8). (Alternatively, equation (10) could be used for Λ.)
6. Place the values of all the dependent variables (T_i, v_i, x_i, Λ) in a "save" vector.
7. Call GEARIB to do one Newton iteration.

8. Check to see whether any mesh interval $(x_{i+1} - x_i)$ has decreased by more than 20% during the Newton iteration. If so, relax the Newton iteration by performing a uniform linear interpolation (for all x_i) between the current and preceding values so that the maximum decrease in any mesh interval is 20%.

9. Evaluate T_i from (16).

10. Solve for v_i recursively using (15).

11. Solve for Λ as in Step (5).

12. Test for convergence, i.e.,

$$\sum_{j=1}^{3N+1} \left(\frac{\Delta Y_j}{YN_j}\right)^2 < [(3N+1)EPS]^2 \quad ?$$

where

i) The summation is over all the dependent variables (T_i, v_i, x_i, Λ).

ii) ΔY_j is the difference between the current and preceding (at the beginning of Step (7)) values of the j^{th} dependent variable.

iii) YN_j is the "normalization" factor for dependent variable j. It is the maximum value of Y_j occurring during all of the preceding iterations, but it must be at least 0.01.

iv) EPS is the same accuracy criterion used for the conventional GEARIB integration.

13. If the iteration has converged go to the beginning of the conventional integration.

14. Test to see if the maximum number of iterations (=50) has been exceeded or if there is an error return from GEARIB. If so, then abort.

15. Go to Step (6).

As mentioned previously, the principal focus of the work described in this report was the study of how the variable mesh adapts itself to the evolving solution. Consequently, the emphasis in the initial condition iteration scheme was on the assurance of convergence rather than on efficiency. This is the reason for the

conservative relaxation criterion of Step 8 and the re-evaluation of the Newton matrix on each iteration. Despite these safety factors, the time required for the determination of initial conditions was generally an insignificant fraction of the total run time.

What about the commonly occurring situation where the initial values of T_i are prescribed at preselected values of x_i? The algebraic equations (15) could be satisfied as before by solving for the v_i recursively. However, equations (8) could not be satisfied. A way out of this difficulty would be to generalize (8) so that the function F would be defined somewhat differently on each mesh interval, but in such a manner that $F_i \to F$ with increasing time. If F specifies the "optimal" distribution of the x_i, then initially and for some time the x_i distribution actually used would be suboptimal, but this cannot be avoided.

All of the F definitions used during this study contain a parameter ε, e.g., equation (9). An easy way to implement the scheme just described would be to preselect Λ and then replace ε by ε_{io} so that (8) would be exactly satisfied on each interval. Then define

$$\varepsilon_i = \varepsilon_{io} e^{-at} + \varepsilon(1 - e^{-at})$$

where a is some positive constant to be determined; probably by experiment.

D. Specification of Finite Difference Mesh Spacing

As noted previously, the principal difficulty of this study is the selection of the function $F(x_i, x_{i+1}, v_i, v_{i+1})$ of equation (8). To motivate this selection consider the spatial discretization error when (3a) is written at the mid point of each mesh interval, i.e., equations (3), (6), ..., (N-4) of system (13).

$$(\dot{T}_i + \dot{T}_{i+1}) - (v_i + v_{i+1})(\dot{x}_i + \dot{x}_{i+1})/2$$

$$= 2\beta(v_{i+1} - v_i)/H_i - \gamma(v_i + v_{i+1})(T_i + T_{i+1})/2 \quad (17)$$

Now expand each term of (17) about the midpoint of the mesh interval. Quantities associated with this midpoint are given a subscript m. Apostrophes denote $\partial/\partial x$.

$$\left(2\dot{T}_m + \frac{H_i^2}{4}\dot{T}_m''\right) - \left(2v_m + \frac{H_i^2}{4}v_m''\right)\left(\dot{x}_m + \frac{H_i^2}{8}\dot{x}_m''\right)$$
$$= 2\beta\left(v_m' + \frac{H_i^2}{24}v_m'''\right) - \gamma\left(2v_m + \frac{H_i^2}{4}v_m''\right)\left(T_m + \frac{H_i^2}{8}T_m''\right) + O(H_i^4) \qquad (18)$$

Thus, the truncation error associated with this spatial discretization is:

$$TE = \frac{H_i^2}{4}\left\{\frac{\beta}{3}v_m''' - \gamma(v_m T_m'' + T_m v_m'') - \dot{T}_m'' + (v_m \dot{x}_m'' + \dot{x}_m v_m'')\right\} + O(H_i^4) \qquad (19)$$

Note that the truncation error is $O(H^2)$ irrespective of any mesh nonuniformity. Contrast this to the classical three point differencing scheme for which the truncation error is $O(H^2)$ only for uniform meshes. Thus, for a preselected truncation error level TE, H_i should be chosen,

$$H_i = \frac{2\sqrt{TE}}{\sqrt{|\{\ \}|}} \qquad (20)$$

There are several reasons why (20) is impractical:

1) To establish a finite difference mesh satisfying (20), it is necessary to fix at least one mesh point, say x_1. Then assuming that { } can be evaluated, (20) can, in principle, be used to specify all the mesh points. However, for an arbitrarily selected value of TE, none of the mesh points so determined would, in general, coincide with the desired right boundary point, x_N. Even if TE could be selected so that one of the mesh points determined from (20) did coincide with x_N initially, this would not continue to hold as the solution evolved in time causing { } to change.

2) If { } passes through zero, H_i from (20) can become arbitrarily large and the denominator of (20) becomes nonanalytic. Both of these effects are troublesome numerically.

3) Many of the terms in { } of (19) cannot be estimated by difference formulas involving values at only two consecutive mesh points. Since a formula similar to (20) is embedded in the differential-algebraic system (13), the number of consecutive mesh points in excess of two used in the formula increases the bandwidth of the Jacobian and, consequently, the computation time.

To circumvent difficulty (1), the numerator of (20) is replaced by an auxiliary variable Λ which is set initially and automatically adjusted as the solution evolves in such a manner that the preselected number of mesh points exactly fits into the specified spatial interval. With this modification we have

$$TE = \Lambda^2/4 \tag{21}$$

Note the change in philosophy, i.e., the number of mesh points is selected, and the spatial truncation error is a consequence of this number. TE also varies in time.

Difficulty (2) is surmounted by replacing the denominator of (20) by $\sqrt{\{\ \}^2 + \varepsilon}$, where ε is a positive constant (typically O(1) in our numerical examples). This change has the effect of making the denominator analytic and preventing the occurrence of arbitrarily large mesh intervals even if $\{\ \} \to 0$. Of course, the smaller ε is chosen the closer to a nonanalytic function $\sqrt{\ }$ becomes, and the more numerical difficulties can be expected.

For difficulty (3) the solution is less satisfactory. Instead of $\sqrt{\{\ \}^2 + \varepsilon}$ which we cannot hope to evaluate within the constraints of our formulation (i.e., minimum bandwidth) we employ a function $F(x_i, x_{i+1}, v_i, v_{i+1})$ (already introduced in equation (8)) which we hope can be selected to mimic the important features of $\sqrt{\{\ \}^2 + \varepsilon}$. Selection of this function is the principal obstacle in the development of an automatic adaptive mesh method.

Implicit in our definition of the function F is that it does not depend on T_i and T_{i+1}. This is not an essential restriction. However, for the example problems considered in this study, T is always in the range [0,1] and so it was felt that T was not a crucial quantity in F. Further, spatial derivatives of T can always be expressed in terms of v.

Another important assumption regarding F is that it not be a function of any of the time derivatives. The reason for this is that GEARIB requires that all time derivatives appear linearly so that the differential-algebraic system can be cast in the form (13). The types of F considered in this study definitely do not have this property.

Of the remaining terms in (19) the only one that can be approximated in terms of values at only two consecutive mesh points is $\gamma v_m T_m''$, i.e.,

$$v_m T_m'' = \left(\frac{v_i + v_{i+1}}{2}\right)\left(\frac{v_{i+1} - v_i}{H_i}\right) = \left(\frac{v_{i+1}^2 - v_i^2}{2H_i}\right) \qquad (22)$$

One could hope that (22) is a reasonable measure of the discretization of the nonlinear term of Burger's equation even though the other $O(H^2)$ contribution, $T_m v_m''$, is neglected.

In typical applications of Burger's equation, the coefficient, β, of the diffusion term on the RHS of (1) is small. For our examples, it never exceeds 0.01. Despite this small coefficient, the diffusion term is important at the "knees" of the wave-like solutions to Burger's equation. For example, see Figure I of the next section and note the sharpness of the knees for larger values of t. As β is reduced these knees become sharper. Although we are not able to estimate v_m''' of (19) required for the truncation error of this diffusion term, we can at least determine when the diffusion term is important relative to the nonlinear term by evaluating the ratio of these two terms, i.e.,

$$R \equiv \frac{\beta^2\left(\frac{v_2 - v_1}{x_2 - x_1}\right)^2}{\gamma^2\left(\frac{v_1 + v_2}{2}\right)^2} \qquad (23)$$

Strictly speaking the denominator of (23) should have a factor $(T_1 + T_2)^2/4$, but this has been omitted for reasons cited previously. When $R << 1$ the diffusion term is not important and (22) may be a good measure of the truncation error corresponding to the RHS of Burger's equation. However, if $R \geq O(1)$, then some function should be devised which forces an increased concentration of mesh points within this region where the diffusion term is important.

The preceding are some of the considerations that guided our selection of the function $F(x_i, x_{i+1}, v_i, v_{i+1})$ for the determination of mesh spacing. Numerical experiments were performed with a variety of such functions. A number of

unexpected phenomena were encountered during these experiments, which are described in the following section.

E. Selection of $F(x_i, x_{i+1}, v_i, v_{i+1})$ and Numerical Results

Most of the numerical results presented in this section are based on the following values:

$$\gamma = 1, \qquad \beta = 0.01, \qquad EPS = 0.0001, \qquad N = 31$$

γ and β are constant parameters in Burger's Equation (1), EPS specifies the accuracy of the time integration in GEARIB, and N is the number of mesh points. This value of EPS is small enough so that the spatial discretization error dominates that of the time integration.

In using the GEARIB integrator, the following were adhered to for all of our numerical work:

1) The Jacobian of the RHS of (13) was evaluated analytically in the user written subroutine PDG. (There is an option in GEARIB to evaluate the Jacobian by numerical differencing, but the analytic approach is generally preferable.)

2) In computing the Newton matrix, the variability of the matrix on the LHS of (13) was explicitly accounted for. Some early runs which did not do this suffered degraded integration performance.

3) To start the integration, the time derivatives of all dependent variables were computed (in order to get a good initial prediction and, consequently, improved corrector convergence properties for the first step.) Computation of these initial derivatives requires differentiation of all algebraic equations of system (13). However, the coding for these differentiations was already present in subroutine PDG, so relatively little additional work was required.

Most of our numerical examples are based on the x-interval [0,2] with the initial condition

$$T(x) = \begin{cases} (1 + \cos(\pi x))/2, & 0 \le x \le 1 \\ 0, & 1 < x < 2 \end{cases} \qquad (24)$$

T(x) has a discontinuity in its second derivative at x=1. Such discontinuities sometimes cause difficulty for nonlinear iteration schemes, especially in the determination of initial conditions. However, derivative discontinuities in initial conditions are not uncommon in practice, so it was considered relevant to use an initial condition such as (24).

For the reasons discussed in Section C, the boundary conditions used are always:

$$v_1 = 0, \qquad T_N = \text{constant} \qquad (25)$$

E.1 Case I

Although not motivated by the truncation error analysis of the preceding section, the first numerical tests used:

$$F(x_i, x_{i+1}, v_i, v_{i+1}) = \sqrt{(v_i + v_{i+1})^2/2 + \varepsilon} \qquad (26a)$$

and

$$F(x_i, x_{i+1}, v_i, v_{i+1}) = \sqrt{(v_i^2 + v_{i+1}^2) + \varepsilon} \qquad (26b)$$

In regions where v_i is small relative to $\sqrt{\varepsilon}$, the mesh spacing is nearly uniform. In regions of steep slopes of the T-profile, the mesh spacing is nearly inversely proportional to the slope.

Included in this section are plots of the numerical results for various interesting and informative cases. We denote these as Case I, Case II, ... In Figures Ia, Ib, and Ic are plotted the results for:

Case I

$$N = 32$$
$$x_N = 1.985$$
$$\varepsilon = 1.0$$

$$(x_{i+1} - x_i) = \frac{\Delta}{\sqrt{(v_i + v_{i+1})^2/2 + \varepsilon}} \qquad (27)$$

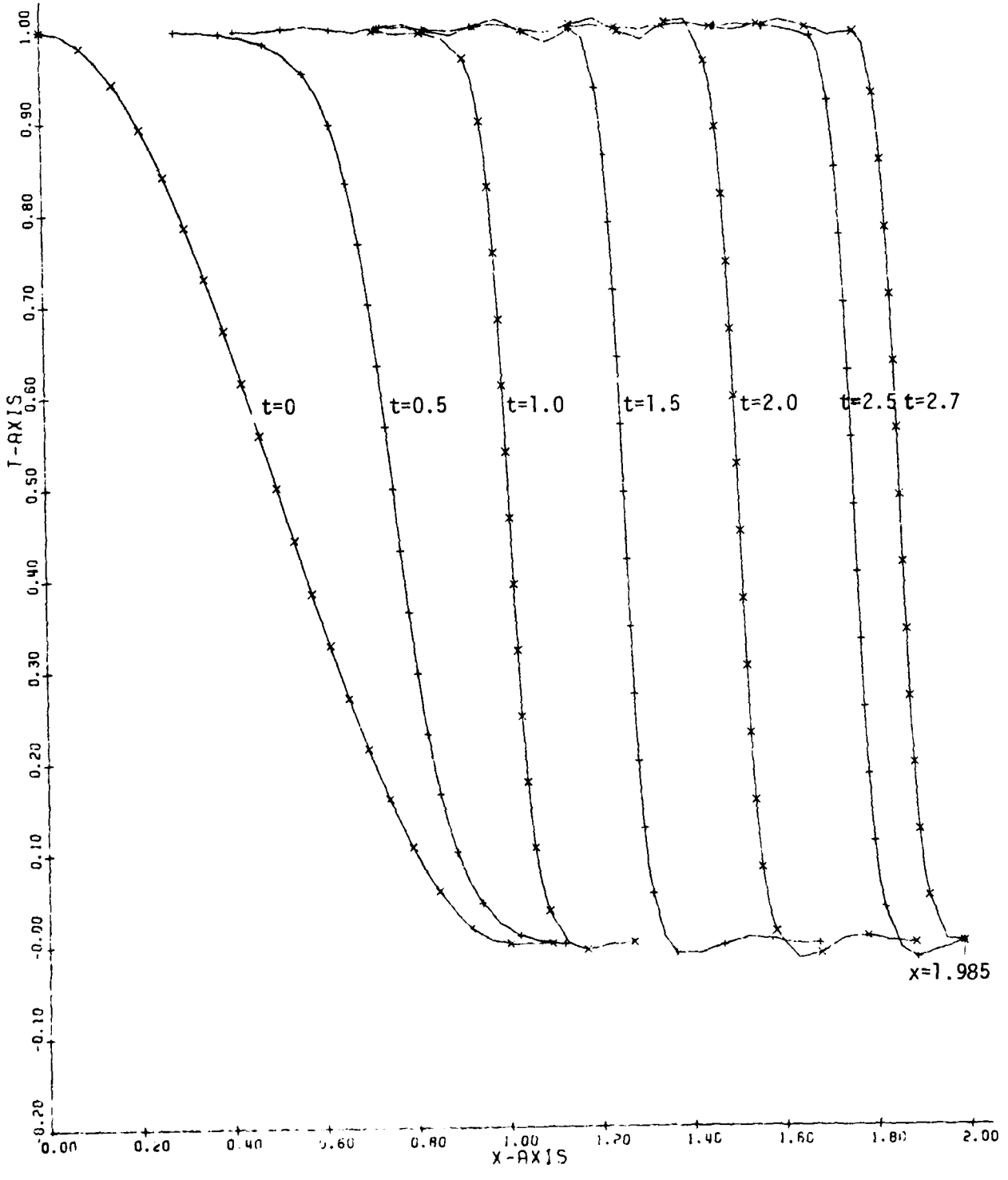


Figure Ia - T Profiles for Case I

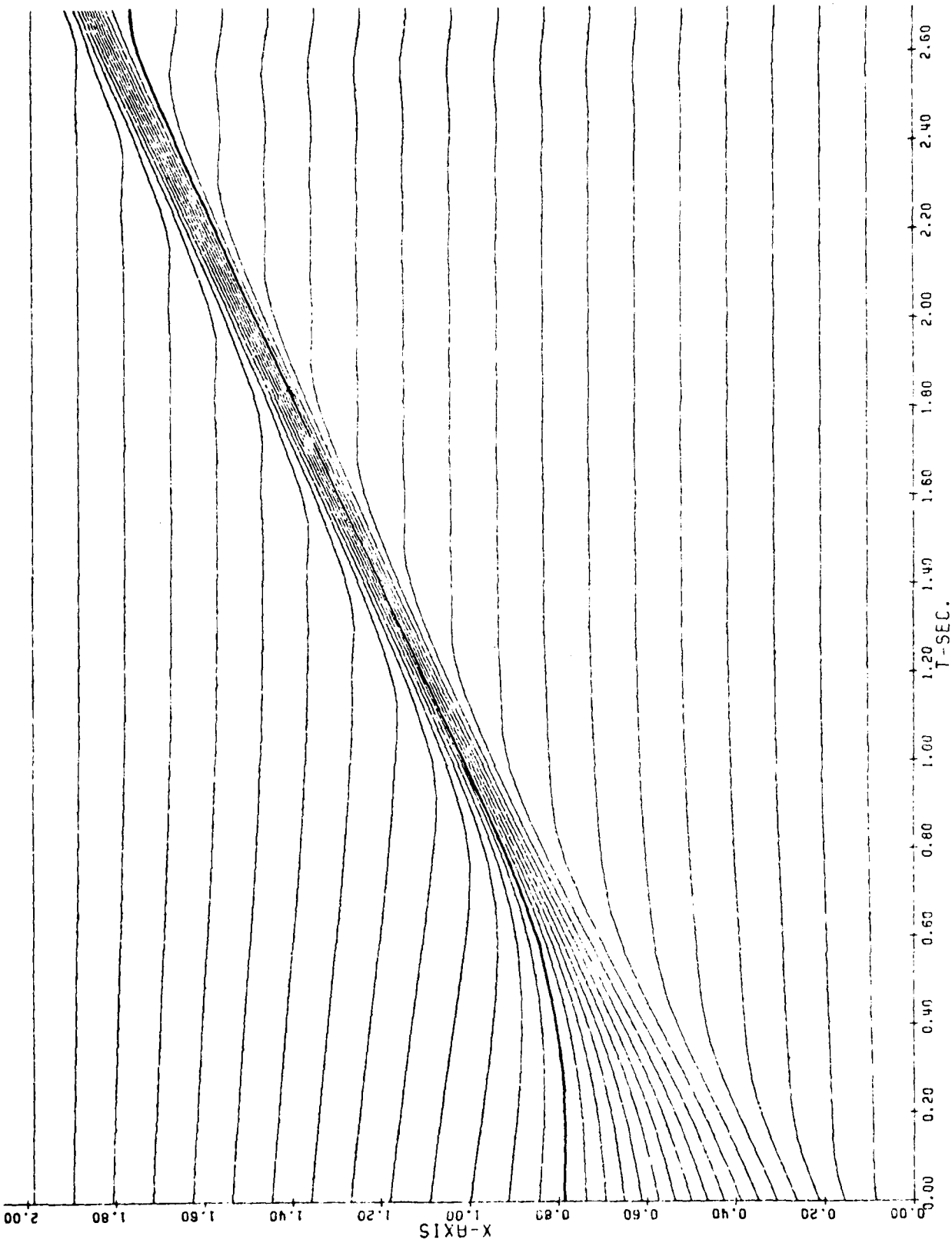


Figure Ib - x-Trajectories for Case I

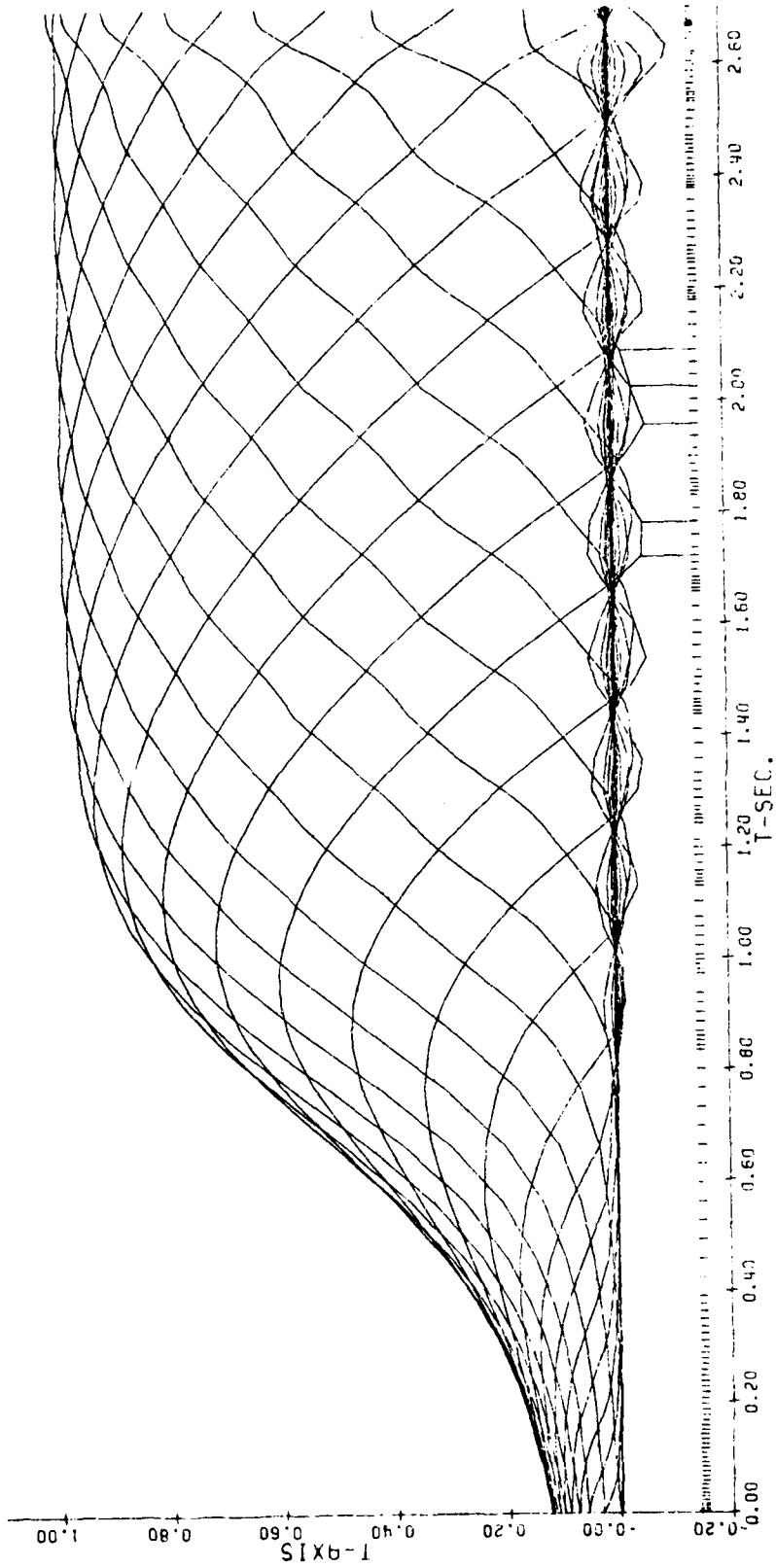


Figure Ic - v-Trajectories and Integration Step Sizes for Case I

The reason for the odd value of x_N is discussed following the presentation of numerical results.

Figure Ia shows the T-profiles at various time intervals up to t = 2.7, which is an arbitrary stopping point. At this time the influence of the right boundary is already strong. (Later examples illustrate the effect of allowing the integration to proceed to "indefinitely large" values of t.) As can be seen from the figure, from t = 1.0 to t = 2.0 (at least) the wave moves to the right practically und storted and with a constant speed. X's and +'s are used on alternate profiles to indicate the solution points (x_i, T_i). In each mesh interval an additional value of T was evaluated at the midpoint using Hermite (cubic) interpolation based on the known quantities $(T_i, T_{i+1}, v_i, v_{i+1})$. Each T-profile was generated by connecting the N points (x_i, T_i) and the (N-1) intermediate points by straight lines. No symbol is used to indicate these intermediate points on the plots. However, in "rough" regions of the profiles these intermediate points are conspicuous by the presence of noticeable slope discontinuities. These slope discontinuities often signal an inappropriate mesh spacing. For example, in Case I the mesh is too coarse in the regions of high curvature near the "knees" of the T profiles, and this is clearly shown in Figure Ia. It is apparent from the figure that the x-interval spanned by a particular T-profile is less than the total solution interval. The plot begins at the first x_i for which

$$(|v_{i+1}| + |v_{i+2}|) > 0.2$$

and it ends at the last x_i for which

$$(|v_{i-2}| + |v_{i-1}|) > 0.2$$

This cutoff criterion (determined experimentally) is intended to prevent excessive overlapping of the T profile plots, which makes it difficult to distinguish certain details.

Another type of plot that was generated for each case is shown in Figure Ib. It shows the trajectories of the x_i in time. This p'ot clearly shows how the region of concentrated mesh points moves along with the wave. It should be noted

that a given mesh point does not remain stationary with respect to the wave. For example, the trajectory of x_{18} is shown in Figure Ib as a heavy line. Initially x_{18} lies in the undisturbed region ahead of the wave. As the wave moves to the right, x_{18} begins to move to the right, but at a speed less than the wave speed. Consequently, x_{18} passes smoothly through the wave region and then slows nearly to a constant asymptotic position behind the crest.

It is clear from this plot that at any time more than half the mesh points lie in a region of nearly constant T. Since such regions of constant T are not of special interest and do not require many mesh points for their description, it is natural to attempt to place more of these mesh points within the wave region. An obvious variant to try is to reduce ε in (27) since the mesh size in constant regions cannot exceed $\Lambda/\sqrt{\varepsilon}$. However, it was found by experiment that reduction of ε much below unity generally resulted in little payoff. The problem became more sensitive numerically resulting in smaller integration step sizes, more iterations per step, and more erratic profiles and trajectories. For example, the slight waviness in the x_i trajectories of Figure Ib was often accentuated when ε was reduced.

A third type of plot generated for each case is shown in Figure Ic. It shows the trajectories of the v_i in time. Since v_i should be negative for the examples being considered in this report, we have plotted $-v_i/|v_{imax}|$, where v_{imax} is the largest value of v_i (over all i and all t) encountered during the integration. From t = 1.0 to t = 2.0 (approximately) the pattern of these v-trajectories is nearly uniform. However, near t = 2.7 the pattern starts to change because of the influence of the right hand boundary. Notice how v_i corresponding to a position ahead of the wave begins to oscillate about zero as the wave approaches, then builds up to its peak value as its mesh point x_i passes through the wave, then decreases and oscillates about zero before reaching a steady zero value.

The tic marks just above the horizontal axis of Figure Ic indicate the integration step sizes that were used. To reduce the memory and CPU requirements of the plotting phase, the trajectories of Figures Ib and Ic were generated by connecting with straight lines the values at integration steps 0, 3, 6, ... This is especially conspicuous in Figure Ic. Several of these straight line segments spanning three integration steps are indicated by vertical lines at the bottom of Figure Ic.

From the plot of step sizes, it is apparent that starting at about t = 0.8, the step size variation has settled into a somewhat regular pattern of increases and decreases. There is no obvious characteristic of the solution which would require such step size changes. More likely it is a deficiency of the GEARIB step size changing criterion. This criterion could probably be tailored to the specific examples being considered in this study in order to reduce this wasteful variation of step size. From t = 0 to t = 0.3, the integration method was first order. At t = 0.3 GEARIB switched (automatically) to a second order method and continued with that order until the end. A few runs similar to Case I were rerun with GEARIB modified so that it would always operate with a first order method. The effect was that the step size variation was somewhat less erratic, but the average step size was less, making the integration less efficient. Beyond this experiment no further effort was expended in tuning GEARIB for the class of problems being studied.

The implicit methods of GEARIB are implemented as an explicit prediction followed by one or more nonlinear corrections. Each of these corrections requires the evaluation of the RHS of (13) and the solution of a linear system. The matrix of this linear system is a linear combination of the matrix on the left of (13) and the Jacobian of the RHS. In GEARIB this matrix is re-evaluated and refactored only when the corrector convergence begins to deteriorate badly. Thus, the number of steps taken is not a reliable measure of the efficiency of the integration. The ratios (NRE/NSTEP) and (NJE/NSTEP) are also crucial. Here NRE is the number of RHS evaluations and NJE is the number of Jacobian evaluations. When GEARIB is operating efficiently, typical values of these ratios might be:

$$\left(\frac{\text{NRE}}{\text{NSTEP}}\right) = 1.5 \text{ to } 2.0, \qquad \left(\frac{\text{NJE}}{\text{NSTEP}}\right) = 0.05 \text{ to } 0.1$$

In Case I these ratios were 4.0 and 0.5, respectively. In some of the other cases to be described later, these ratios improved but never reached the hoped-for levels.

Case I was repeated using (26b) and modifying (13) to use sums of products for terms such as $v\dot{x}$ and vT (instead of products of sums). This resulted in an improvement in the smoothness of the T-profiles and the v-trajectories and the

integration statistics were better. Based on these results, this modification of the Case I formulation seems preferable. However, the Case I and "modified Case I" results are based on a somewhat artificial example. In both instances, x_N was adjusted so that when the initial x_i were solved for (to satisfy (27)), one of the interior x_i fell exactly at x=1. Then when the initial v_i were evaluated based on the T_i obtained from (24), $v_i = 0$ for $x_i \geq 1$. These smooth initial conditions were apparently the reason for the superior performance of the modified Case I. Later "more realistic" cases were run where the right boundary point was set to 2 a priori. Subsequent computation of the initial x_i resulted in x = 1 falling in the interior of a mesh interval with the consequence that $v_{i+1} = - v_i$ for $x_i > 1$. These rougher initial conditions had little effect on Case I but the performance of the modified formulation was significantly degraded. Based on these experiments it was felt that the product of sums formulation was preferable. All of the numerical results presented in this section are for "product of sums" formulations.

One interesting difference between Case I and the modified formulation was in the speed of the wave as it traveled practically without distortion (say between t = 1.0 and t = 2.0). Theory (see [7]) says that the wave speed should be $\gamma/2$ (= 0.5 in our case). For Case I, the T-profiles obey this to as close as can be measured on the plot. For the modified formulation, the wave speed is 0.48. When translated appropriately, the profiles themselves are practically indistinguishable in the regions of steep slope. The reason for this difference in wave speeds might be expected to lie in the discretization of the nonlinear term γvT on the RHS of (3a). For Case I this is:

$$\gamma\left(\frac{v_i + v_{i+1}}{2}\right)\left(\frac{T_i + T_{i+1}}{2}\right) \tag{28a}$$

whereas for the modified Case I it is:

$$\gamma(v_i T_i + v_{i+1} T_{i+1})/2 \tag{28b}$$

If (28a) were consistently larger than (28b), this might explain the larger wave speed corresponding to (28a). However, it is easy to show that (28b) is larger as long as $(v_i - v_{i+1}) > 0$ which is over the upper half of the wave, approximately. Thus, the reason for the difference in wave speeds is not obvious.

E.2 Case II

The oscillations at the crest and trough of the T-profiles of Figure I.1 clearly indicate the inadequacy of the mesh spacing formula (27). More mesh points are needed in the regions of high curvature at the crest and trough. Probably fewer points are needed in the steep portion of the wave. To get this increased concentration in regions of high curvature formula (22) was used in conjunction with (19) (neglecting all terms except $v_m T_m''$).

Case II

$$N = 31$$

$$x_N = 2.0$$

$$\varepsilon = 1.0$$

$$(x_{i+1} - x_i) = \frac{\Lambda}{\sqrt[4]{\left(\frac{v_i + v_{i+1}}{2}\right)^2 \left(\frac{v_{i+1} - v_i}{x_{i+1} - x_i}\right)^2 + \varepsilon}} \qquad (29)$$

The initial T_i again satisfied the formula (24). The boundary conditions were in accordance with (25). Figure IIa shows the T-profiles for t = 0, 0.5, 0.6479. The integration "hung up" at t = 0.6479 for reasons to be discussed shortly. Because of a plotter malfunction, the profiles between x = 0 and x = 0.2 are distorted. The initial profile is identical to that of Figure Ia. Figures IIb and IIc show the x_i and v_i trajectories and the integration step sizes.

The difficulty with using formula (29) to specify the mesh spacing can be seen from Figure IIb. The initial mesh spacing seems to be preferable to that of Figure Ib, since the mesh is more concentrated in regions where the curvature of the T-profile is large. As the wave moves to the right, the mesh intervals $(x_{i+1} - x_i)$ corresponding to small i reach their maximum value ($= \Lambda/\sqrt[4]{\varepsilon}$) and so these mesh points can no longer follow the wave. In order to obtain the required concentration of mesh points in the region of high curvature near the crest, the following pattern occurs:

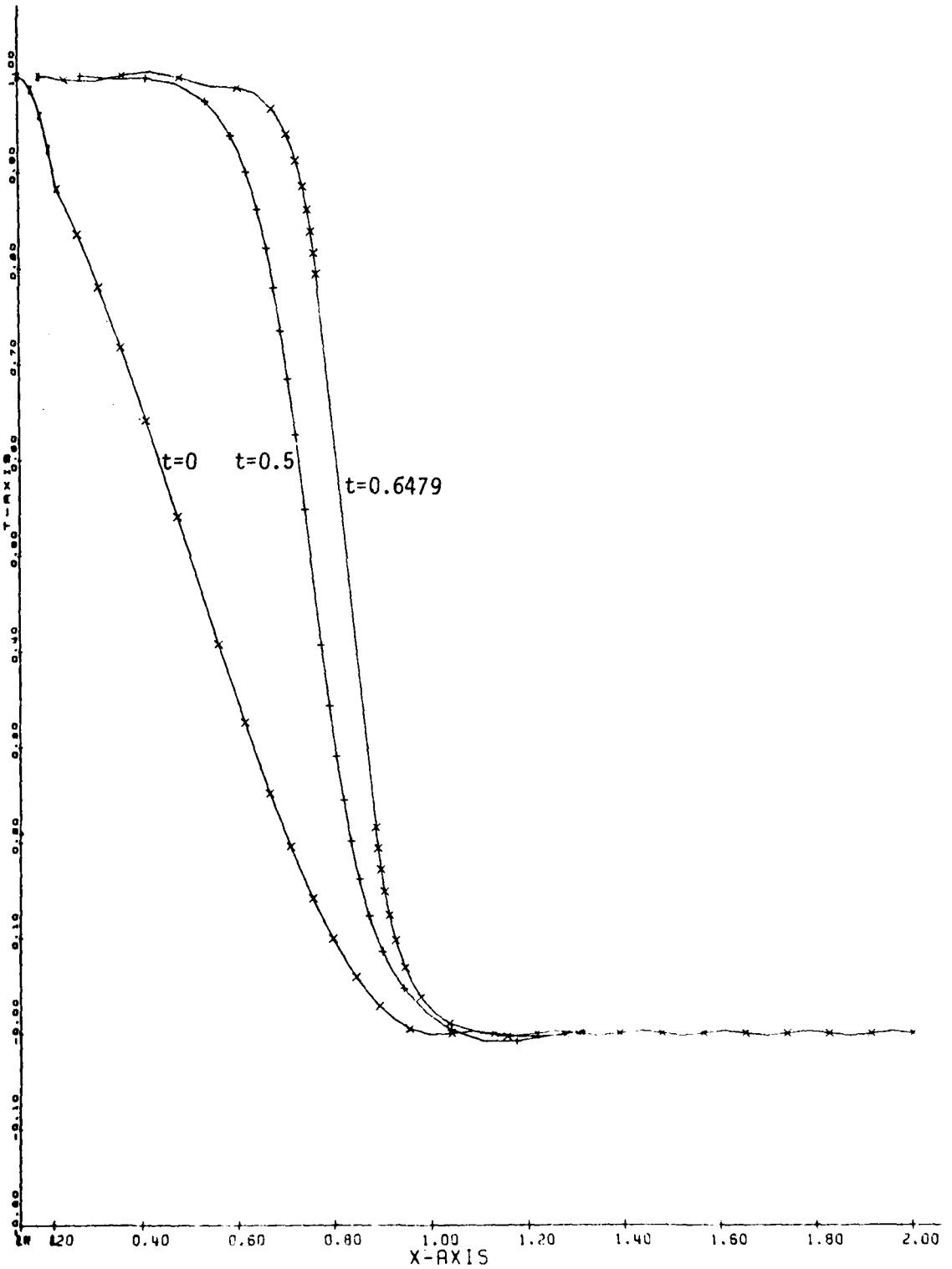


Figure IIa - T-Profiles for Case II

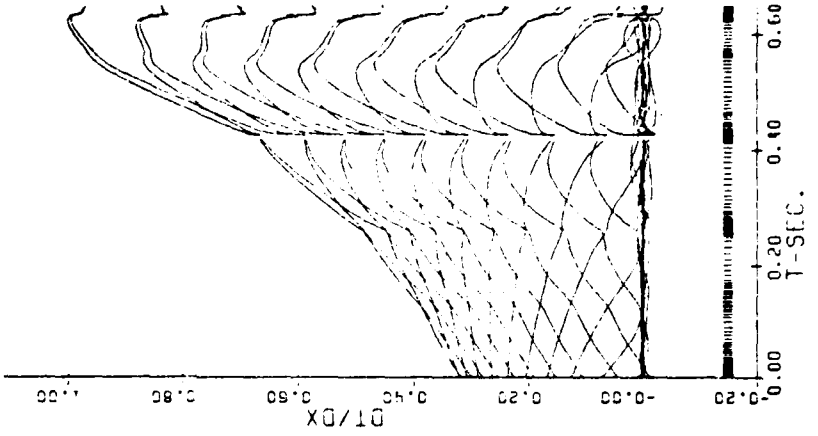


Figure IIc - v-Trajectories & Integration Step Sizes for Case II

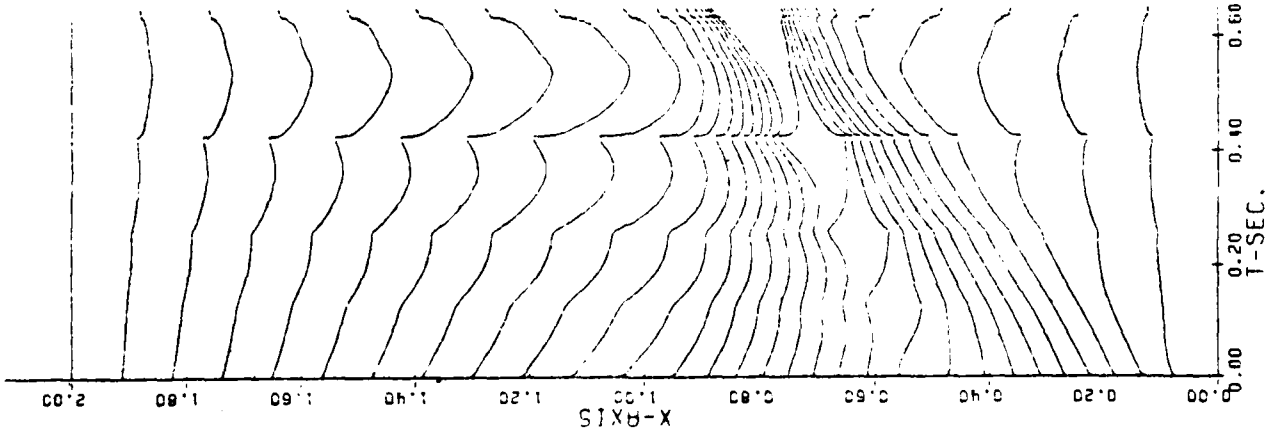


Figure IIb - x-Trajectories for Case II

1) A mesh point moves out of the region of coarse mesh in the middle of wave toward the crest. Since the lower part of the wave is nearly a mirror image of upper part, another mesh point moves out of the middle of the wave toward the trough.

2) This continues until the mesh interval at the center of the wave reaches its maximum value ($= \Lambda/\sqrt[4]{\varepsilon}$). This situation is characterized by $v_i = v_{i+1}$, where x_i and x_{i+1} are the mesh points in the steepest part of the wave. This is readily seen from Figure IIc where the v_i trajectories for the two largest values of $|v_i|$ intersect at various values of t. (The trajectories corresponding to the other large values of $|v_i|$ also intersect in pairs at the same t-points.)

3) Now the only way to achieve the required mesh concentration at the crest is for a mesh point to cross over from the trough to the crest. As the wave steepens, these crossovers become more and more violent creating disturbances throughout the field. These crossovers cause difficulty for the GEARIB integrator as can be seen from the step sizes shown on Figure IIc.

Figure IIb shows that this pattern has been repeated four times. At the end of the run (t = 0.6479) the mesh interval at the center of the wave has reached its maximum and it is time for another crossover to begin. However, the beginning of this crossover is accompanied by such violent changes in the trajectories that the integrator continually failed to take another successful step even though the step was retried with ever smaller step sizes. The final attempt was with a step 10 orders of magnitude smaller than that of the first attempt. Although it cannot be discerned from Figure IIc, the GEARIB integrator was just barely able to negotiate the crossover beginning near t = 0.42. The final T-profile of Figure IIa also illustrates the "extreme" distribution of mesh points and the v-oscillations associated with this distribution.

Several variations of the mesh spacing formula (29) were attempted. One was:

$$(x_{i+1} - x_i) = \frac{\Lambda}{\sqrt[4]{\left(\frac{v_{i+1} - v_i}{x_{i+1} - x_i}\right)^2 + \varepsilon}} \qquad (30)$$

This formula was used with both $\varepsilon = 1.0$ and $\varepsilon = 10$. After the initial conditions converged, the integrator was unable to take even one successful step. The reason for this appears to be the following:

The initial conditions on v in the region of constant T ($x = [1,2]$) are such that $v_i = v_{i+1}$. The magnitude of these "spurious" v_i are typically small, e.g., $|v_i| = O(0.05)$. However, the difference quotient in (30) based on these v_i is $O(1)$, and so Λ (from formula 10) is significantly affected by these small v-oscillations. Now the effect of the diffusive term in Burger's equation is to smooth out these high frequency oscillations. Because there are so many of these oscillations, the cumulative effect of their being smoothed out is to cause a rapid reduction in Λ, which in turn forces a rapid change in all the mesh points through formula (30). The integrator is unable to cope with these violent changes.

There are several ways out of this v-oscillation difficulty:

1) "Doctor" the problem so that the initial conditions do not contain these spurious v-oscillations. For example, this was accomplished in Case I by adjusting the right boundary, x_N, so that a mesh point fell exactly at $x = 1$. Such doctoring is not in the spirit of developing a general purpose numerical technique so nothing further was done in this direction.

2) Reduce the interval of interest to $x = [0,1]$. The initial conditions now have no region of constant T and consequently no region of v-oscillation. This problem was run with $\varepsilon = 10$. There was no longer any difficulty getting the integration started. However, the integration "hung up" at $t = 0.5779$ because of a crossover effect similar to that discussed in conjunction with Case II. Due to the shortened x-interval, the right boundary affected the wave from the beginning and so the upper and lower parts of the wave were not mirror images for $t>0$. Consequently, the T-profiles and the x_i and v_i trajectories are somewhat different than those of Figure II. The basic difficulty is still the same, i.e., as the lower part of the wave straightens out (e.g., like the last profile of Figure Ia) fewer mesh points are needed there, and so from time to time a mesh point must cross the steep portion of the wave and into the crest to compensate for the mesh points that have not been able to follow the wave. As the wave becomes steeper, this crossover becomes more and more

violent and finally the integrator quits. Of course, changing the x-interval from [0,2] to [0,1] in order to get the integration started is an artifice which is in general unacceptable.

3) A more satisfactory approach is to modify the mesh spacing formula to nullify the effect of the spurious v-oscillations. In fact, this has already been done in Case II where the mesh spacing formula (29) contains the factor $(v_i + v_{i+1})$. Other variations, all containing this crucial factor, are discussed in the following:

One mesh spacing formula for which a number of tests were run is:

$$(x_{i+1} - x_i) = \frac{\Lambda}{\sqrt[4]{\left(\frac{v_{i+1} - v_i}{x_{i+1} - x_i}\right)^2 \left[\frac{(v_i + v_{i+1})^2}{2(v_i^2 + v_{i+1}^2 + 10^{-30})}\right]} + \varepsilon} \qquad (31)$$

In regions where T varies significantly, the ratio [ ] in (31) is close to unity and so (31) is nearly equivalent to (30). In regions where T is constant, $v_i = -v_{i+1}$, and (31) is equivalent to (29).

When T is not constant formula (31) (in comparison to formula (29) of Case II) has the effect of putting a greater concentration of mesh points in regions where v is small (e.g., near the crest and the trough) and a lesser concentration where v is large (e.g., in the middle of the wave). The numerical results using (31) were similar to Case II. However, (31) produced smoother T-profiles and x and v trajectories, and the integration was about 25% more efficient (for the same value of t). Equation (31) allowed one more crossover before becoming hung up (at t = 0.86). Another run using (31) with $\varepsilon = 10$ (instead of $\varepsilon = 1$) produced very similar results but was about 25% more efficient. Still another run was made with (31) and $\varepsilon = 10$ but with the number of mesh points, N = 51 (instead of 31). Crossovers were more frequent, but less violent. However, the integration eventually hung up at t = 1.05 (at the beginning of the 13th crossover) for the same reason as before. The integration was about 25% less efficient (in terms of the number of corrector iterations and Jacobian re-evaluations) than the N = 31, $\varepsilon = 10$ case. The T-profiles were somewhat smoother as would be expected from the increased number of mesh points.

E.3 Case III

In this section we present numerical results for a case somewhat similar to the case discussed at the end of the preceding section.

Case III

$$N = 51$$

$$x_N = 2.0$$

$$\varepsilon = 10.0$$

$$(x_{i+1} - x_i) = \frac{\Lambda}{\sqrt{\left(\frac{v_{i+1} - v_i}{x_{i+1} - x_i}\right)^2 \left[\frac{(v_i + v_{i+1})^2}{2(v_i^2 + v_{i+1}^2 + 10^{-30})}\right] + \varepsilon}} \qquad (32)$$

The difference between Case III and that discussed at the end of the preceding section is the use of $\sqrt{\ }$ in (32) instead of $\sqrt[4]{\ }$. The numerical results are presented in Figures IIIa, IIIb, and IIIc. The use of $\sqrt{\ }$ results in quite different looking x and v trajectories. There is still the crossover effect which occurs when the mesh spacing at the center of the wave reaches its maximum ($= \Lambda/\sqrt{\varepsilon}$). Now for Case II and the other similar cases discussed in the preceding section, there was a sudden change in the x and v trajectories at such maximum points (see Figures IIb and IIc). In contrast, for the present case this maximum mesh spacing decreases gradually and the trajectories are smooth. Then suddenly the mesh point crossover takes place and disturbances are propagated throughout the field. As expected, the integration step size is greatly reduced at such points.

From Figure IIIb it can be seen that the integration hangs up at a time when the trajectories are smooth, in contrast to the cases previously discussed, e.g., Figure IIb. The reason that the integration stopped in Case III was that two mesh points became equal to six figures making the problem numerically too sensitive for the integration to continue. The effect of using $\sqrt{\ }$ (instead of $\sqrt[4]{\ }$) in the mesh spacing formula is to force the mesh points even closer together

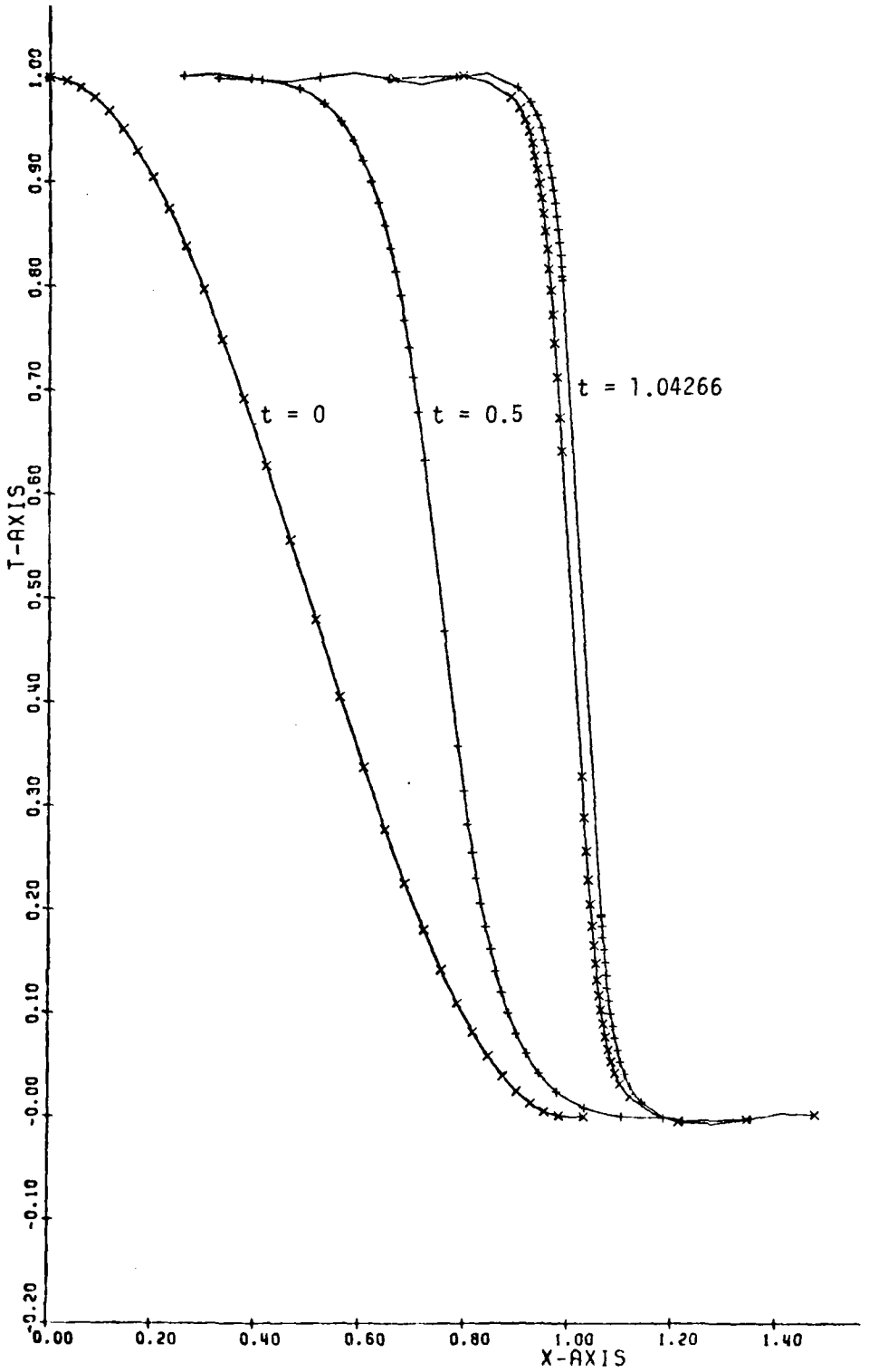


Figure IIIa - T-Profiles for Case III

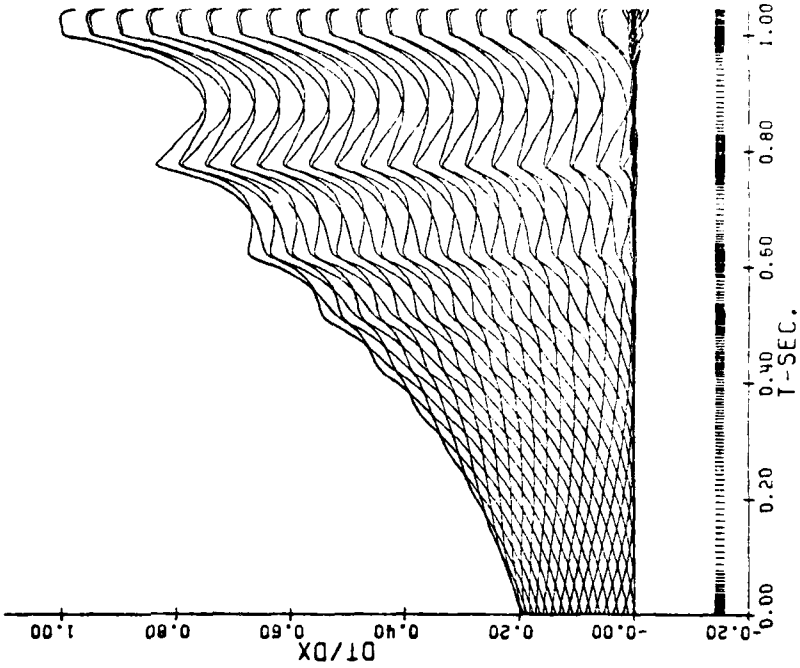


Figure IIIc - v-Trajectories and Integration Step Sizes for Case III

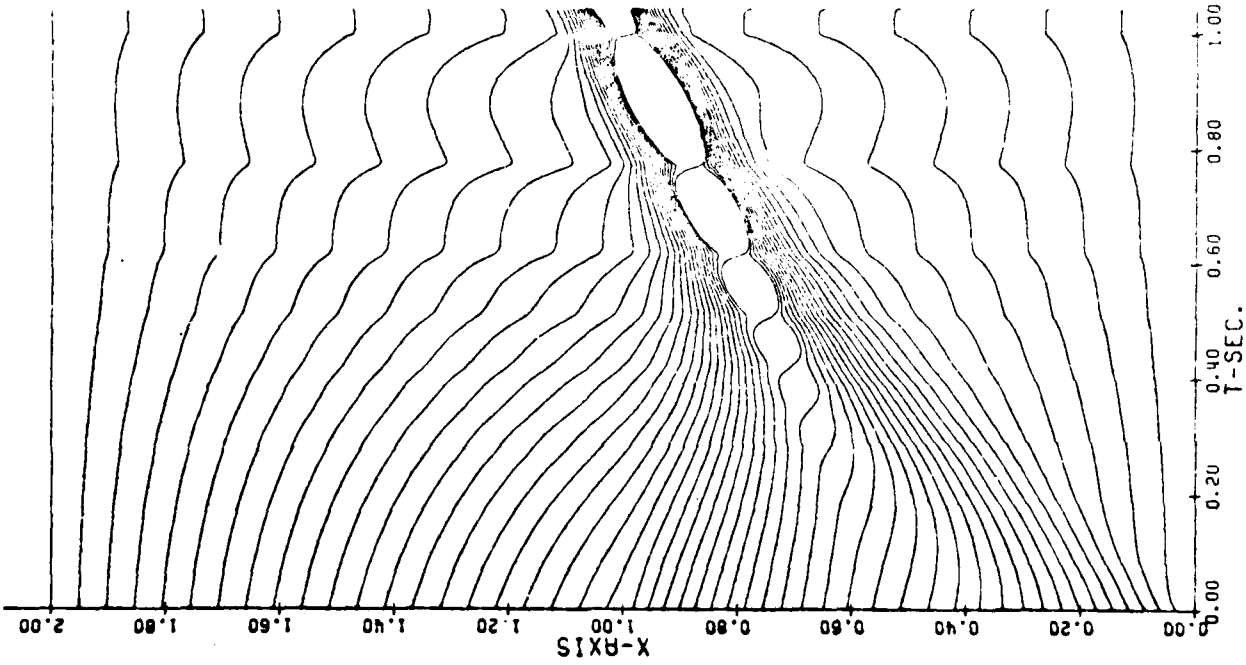


Figure IIIb - x-Trajectories for Case III

in regions of high curvature. However, coincidence of mesh points to six figures is almost certainly a numerical problem. One way to avert such difficulties is to modify (32) so as to place a lower bound on $(x_{i+1} - x_i)$, e.g.,

$$(x_{i+1} - x_i) = \Lambda\left(\frac{1}{\sqrt{\ }} + \delta\right) \qquad (33)$$

where δ is related to the minimum permissible mesh spacing. This approach was not pursued, however, since the $\sqrt{\ }$ formula seemed to be preferable (in terms of integration efficiency and smoothness of the T-profiles) to formula (32) even before excessively small mesh sizes became a problem.

E.4 Case IV

Problems arose in the examples of the preceding subsections due to the fact that as the wave moved to the right, additional mesh points were needed behind the wave. These mesh points had to be supplied from those initially ahead of the wave, and the only way they could get to where they were needed was to pass through the wave. As the wave became steeper, this passage became more and more difficult and finally prevented the integration from continuing.

This suggests adding mesh points from the left boundary, as required, making it unnecessary for mesh points to pass through the wave. To do this "properly" would require:

1) Establishing the criterion by which a new mesh point could be admitted from the left boundary. Most likely this would be based on how near $(x_2 - x_1)$ is to its maximum allowable value. It would be necessary to monitor for such a condition frequently.

2) Since $(x_{i+1} - x_i)$ appears in the denominator of some of the terms of the governing differential-algebraic system (13) it would be necessary to develop special formulas to allow the newly introduced mesh interval to start off with a zero value. Alternatively, the new mesh interval might start off with a finite value (say by halving the interval $(x_2 - x_1)$ that existed just prior to the admission of the new mesh point). However, this would require the specification of new values of v and T at this new interior field point such that all of the algebraic equations are satisfied.

3) Shift all the x_i, v_i, T_i ($i = 2,3,\ldots,N$) and Λ of the dependent variable vector, add the x, v, and T values for the new mesh point, and then restart the integration.

Although this approach is probably possible, it would require a considerable amount of analysis and reprogramming to accomplish it. A simple alternative approach is to force x_1 to move to the right in time. This is easily accomplished by prescribing $f_1(t)$ on the right of the second equation of (13) to be a positive monotonic function of t. As noted previously, theory says that the wave-like solutions of Burger's equation travel at a speed $\gamma/2$ (= 0.5 in the present case). We then define:

Case IV

$$N = 31$$

$$x_1 = t/2$$

$$x_N = 2.0$$

$$\varepsilon = 1.0$$

$$(x_{i+1} - x_i) = \frac{\Lambda}{\sqrt[4]{\left(\frac{v_{i+1} - v_i}{x_{i+1} - x_i}\right)^2\left[\frac{(v_i + v_{i+1})^2}{2(v_i^2 + v_{i+1}^2 + 10^{-30})}\right]} + \varepsilon} \qquad (34)$$

Note that (34) is identical to formula (31) discussed in Section E.2. The numerical results are illustrated in Figures IVa, IVb, and IVc. From Figure IVb, it can be seen that "crossovers" still occur. However, the presence of the moving left boundary reduces their frequency. When the identical case was run with a fixed left boundary, the integration "hung up" at t = 0.86 (near the end of the fifth crossover). The present case hung up at t = 1.10 (near the end of the third crossover). Further, the integration was about 50% more efficient (for the same values of t) for the present case.

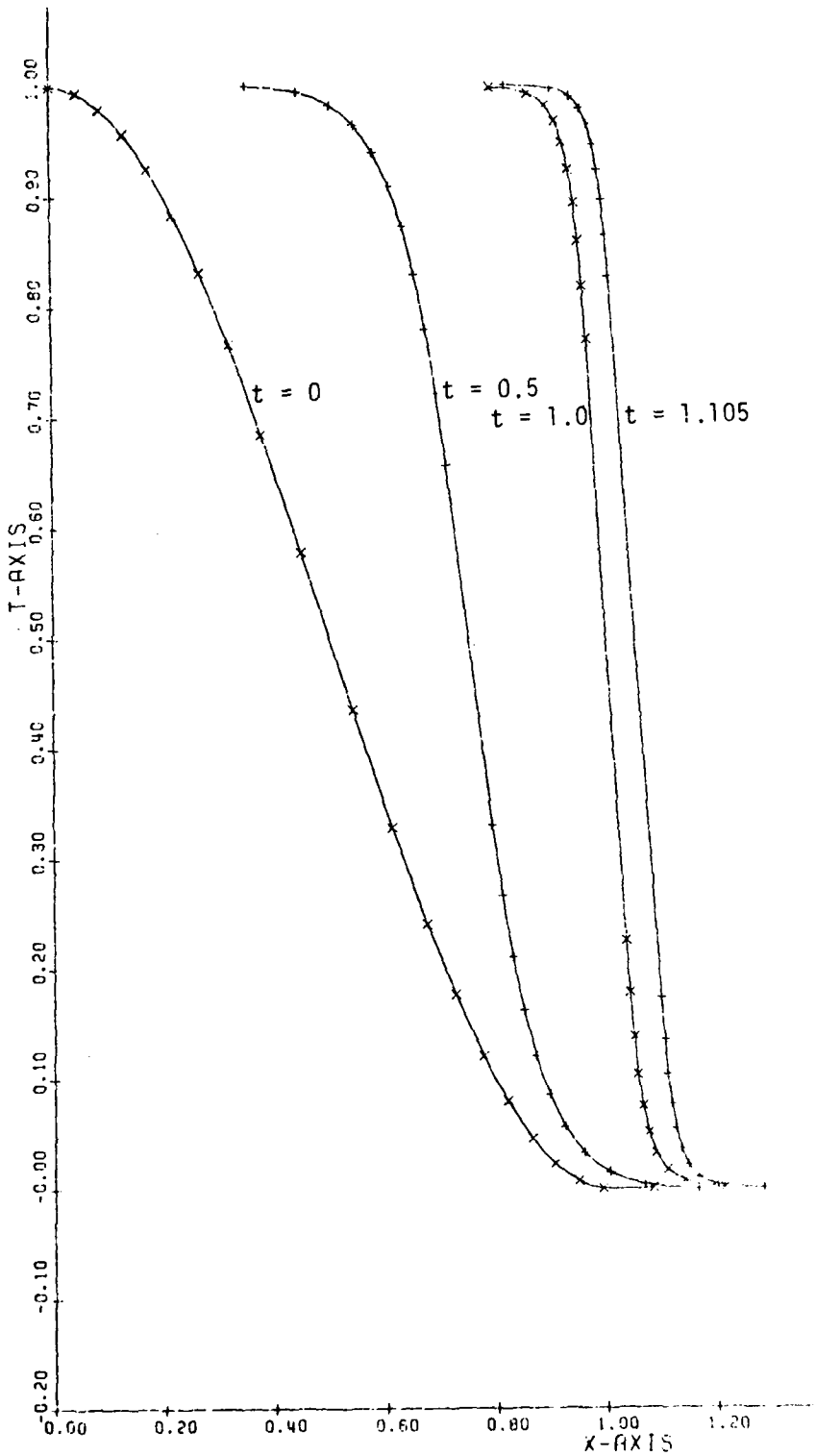


Figure IVa - T-Profiles for Case IV

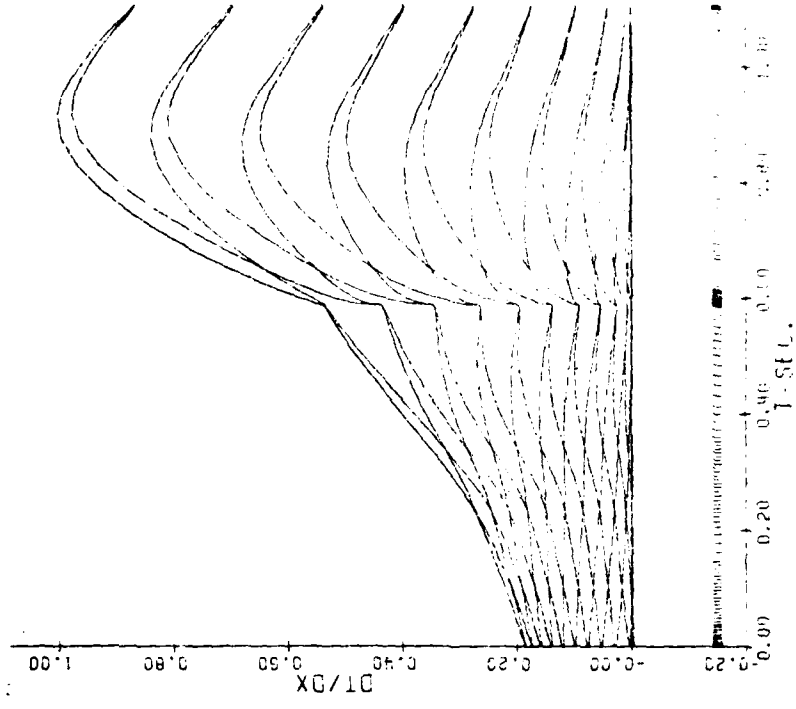


Figure IVc - v-Trajectories and Integration Step Sizes for Case IV

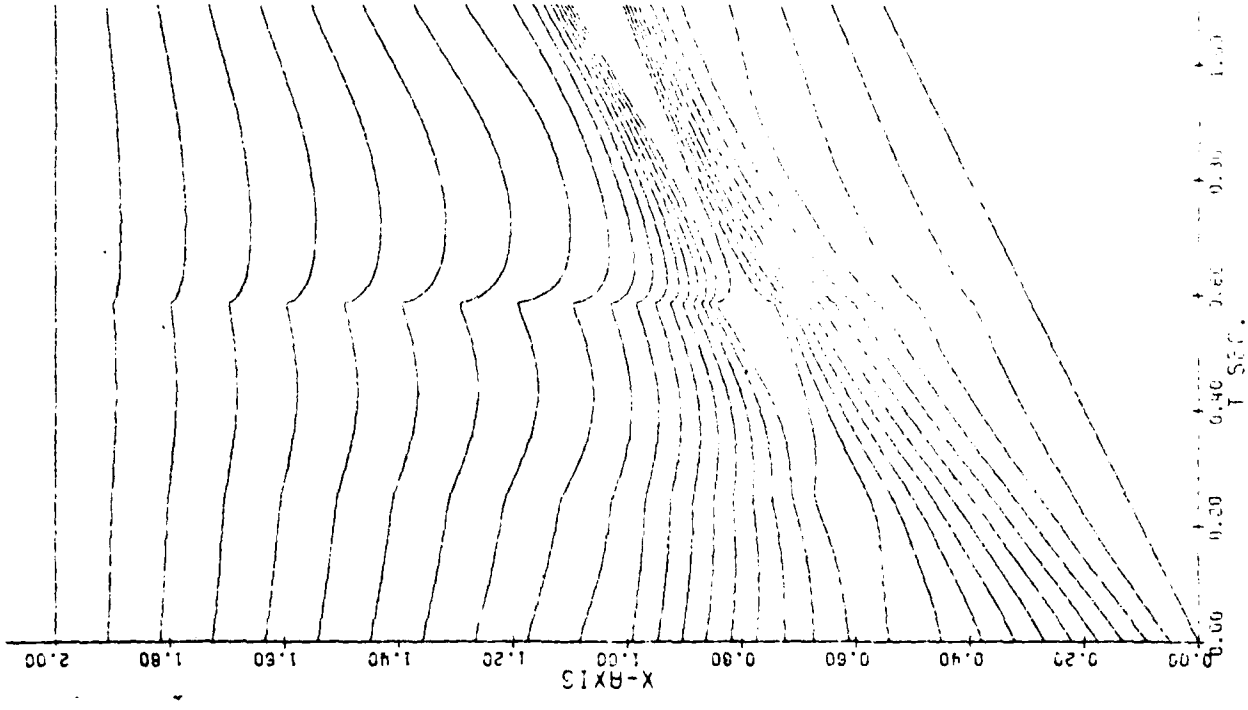


Figure IVb - x-Trajectories for Case IV

E.5 Case V

Forcing the left boundary to move with the wave speed reduced the crossover effect and allowed the integration to proceed further. However, the problem is still essentially the same as before, i.e., as the wave moves to the right there is an excessive number of mesh points ahead of the wave (relative to those behind the wave). An obvious remedy for this is to remove mesh points from the right boundary as they are no longer needed. Such removal of mesh points entails some of the same difficulties cited in Section E.4 when mesh points are added through the left boundary. Generally though, removal of mesh points is somewhat easier than addition. This is discussed further in the next section. For the present, we simulate the removal of mesh points by forcing the right boundary to move with the wave speed.

Case V

$$N = 31$$

$$x_1 = t/2$$

$$x_N = 2.0 + t/2$$

$$\varepsilon = 1.0$$

$(x_{i+1} - x_i)$: Formula (34), Same as Class IV

The numerical results are shown in Figures Va, Vb, and Vc. Because of a plotter malfunction, the beginning (near x = 0) of Figure Va is slightly distorted. In Figures Vb and Vc, the plots end near t = 2.7 even though the integration went to at least t = 2.9. This is due to a plot program idiosyncracy which does not plot the last integration point, which was reached by a large integration step ($\Delta t > 0.2$). Also, the uniform region above x = 2.2 on Figure Vb has been trimmed off.

From the plots it is apparent that the use of boundary points which move at the wave speed has eliminated problems associated with crossovers. The mesh now moves along with the wave and all dependent variables behave either as constants or linearly in time allowing large integration steps to be taken. The large reduction in step size near t = 1.10 does not appear to be related to any properties of the solution and so should be attributed to a numerical integration fluk

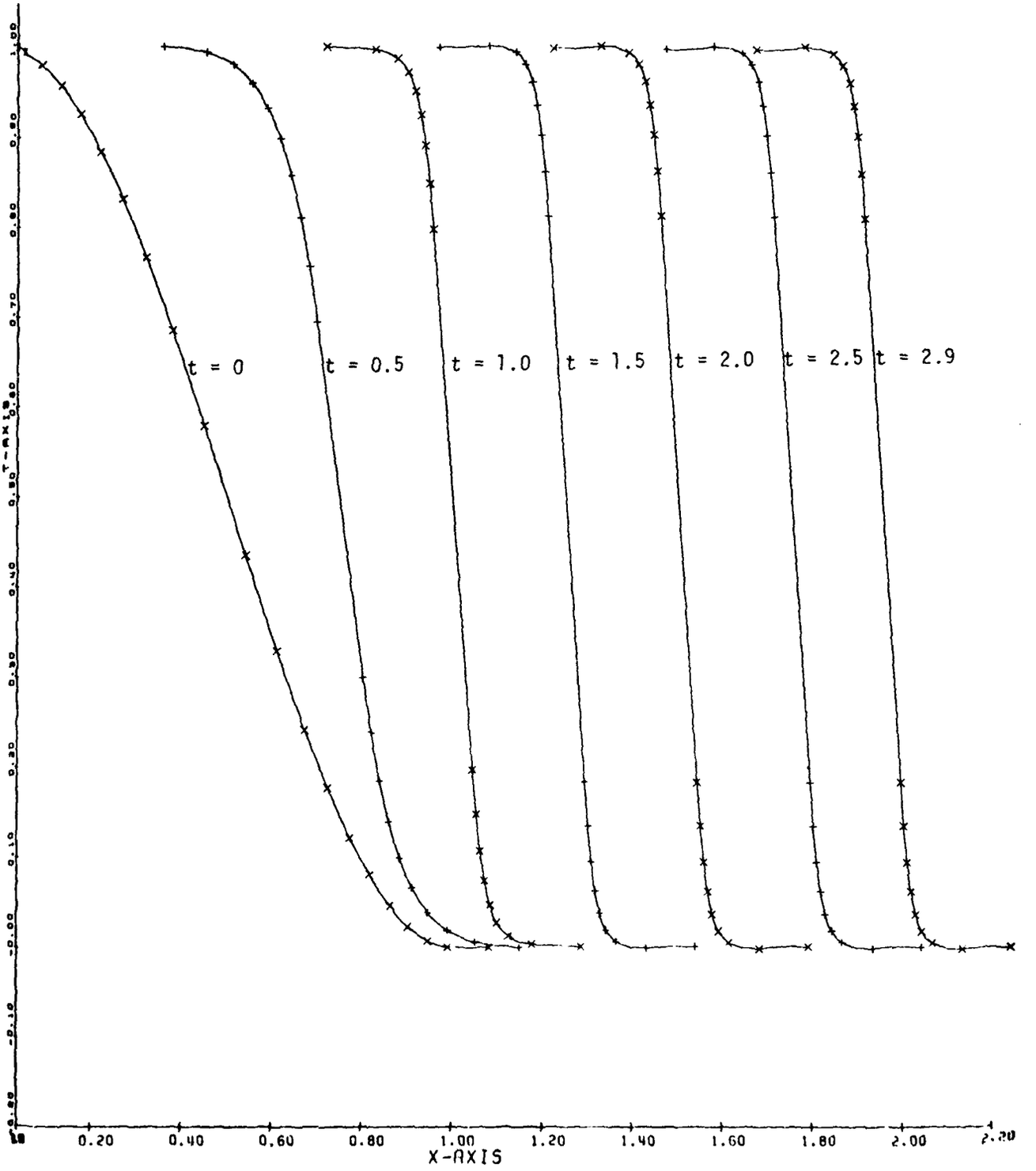


Figure Va - T-Profiles for Case V

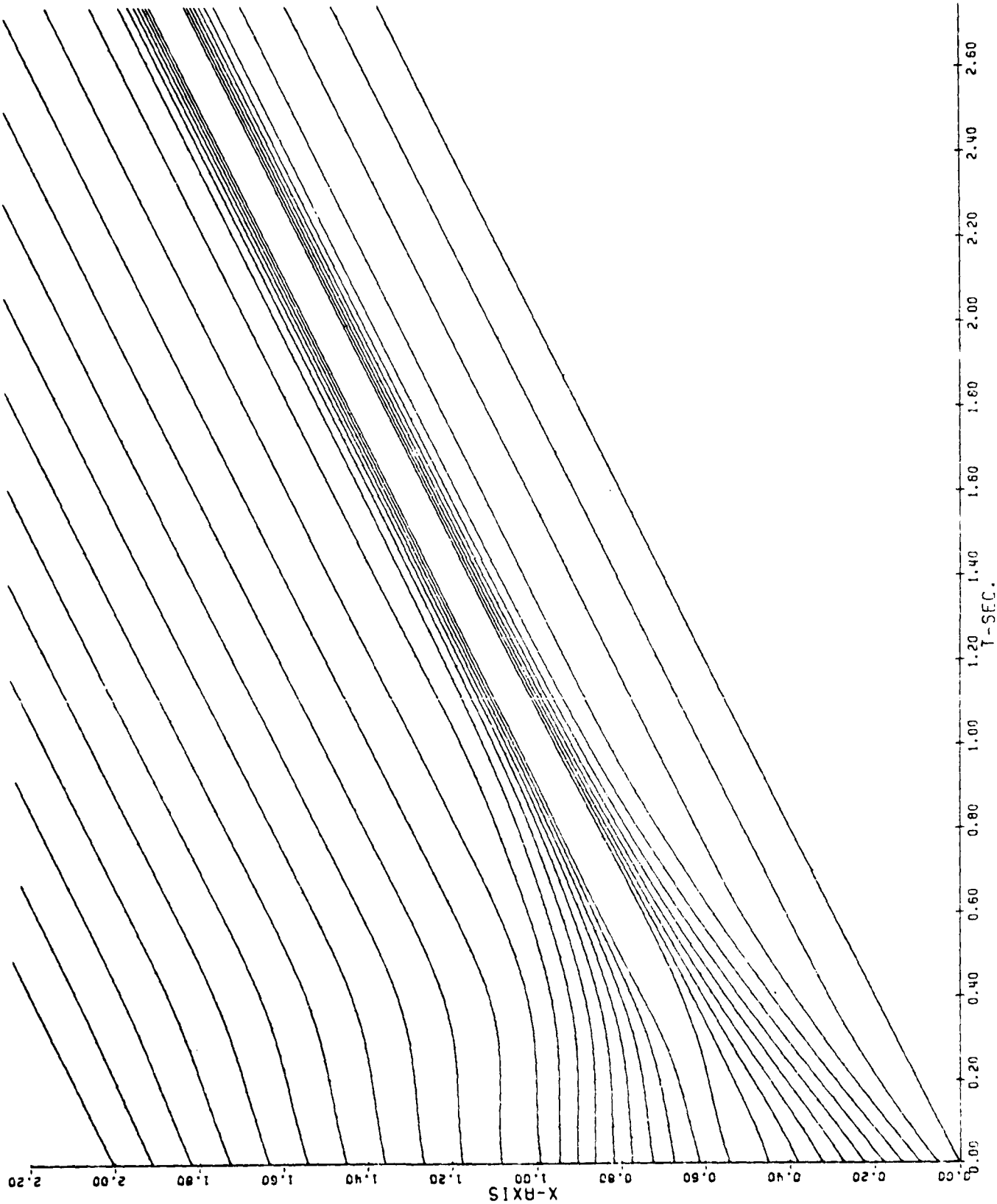


Figure Vb - x-Trajectories for Case V

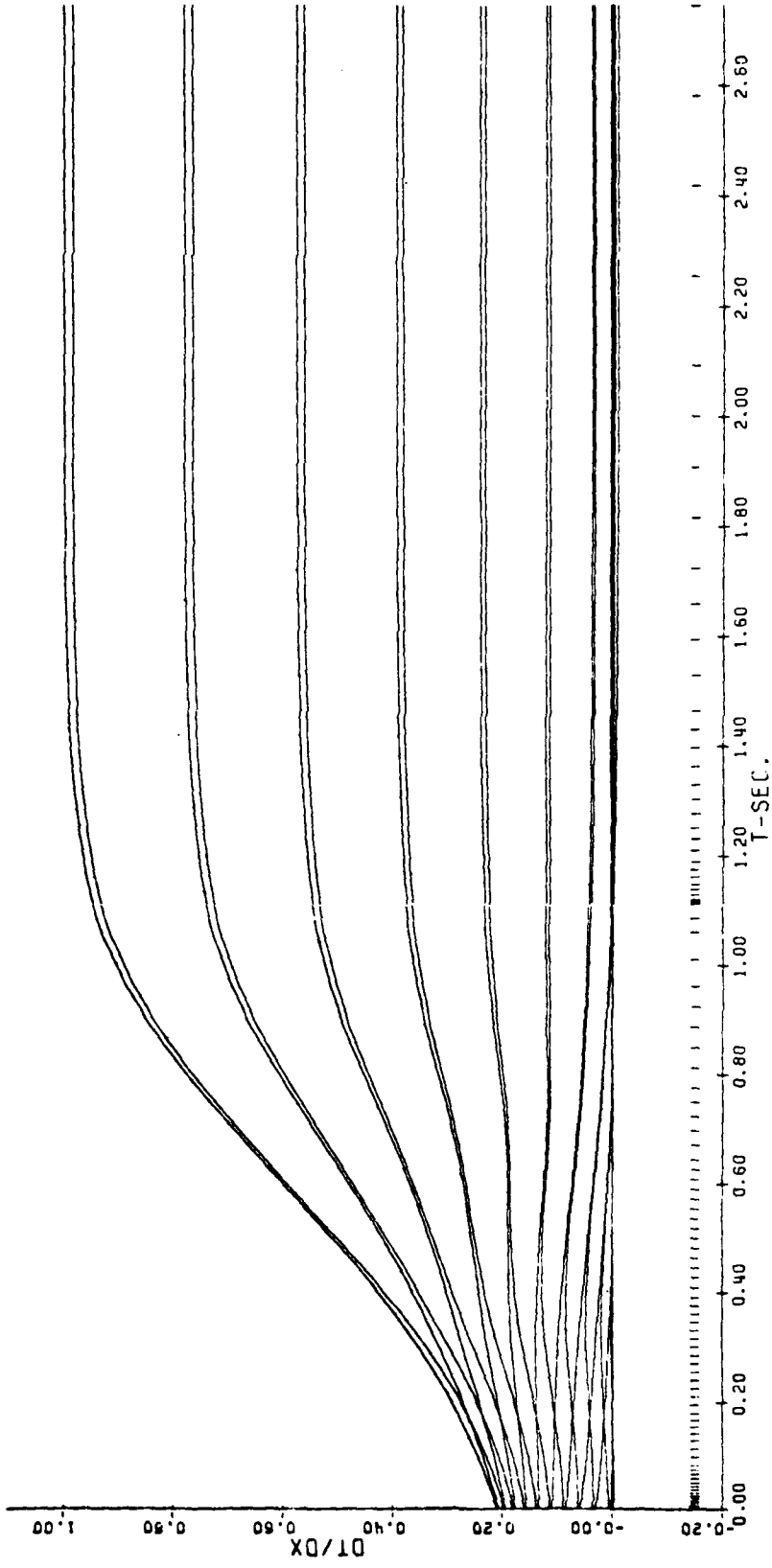


Figure Vc - v-Trajectories and Integration Step Sizes for Case V

E.6 Case VI

Although the use of moving boundaries in Case V allowed the integration to proceed smoothly, it is not a satisfactory solution for the following reasons:

1) Knowledge of the direction and speed of the wave is required. In more general problems, e.g., gas dynamics, this information would not generally be known.

2) The boundary conditions on v and T are applied at the moving boundaries instead of at the original fixed boundaries. Thus, the problem actually solved is not the problem originally posed. For Case V, there is probably little difference between the actual problem and the problem solved until the wave begins to be affected by the downstream boundary (at about t = 2.5 according to Figure Ia).

To overcome objections (1) and (2) but still retain the desirable properties of Case V, the following modification of system (13) was considered:

1) For the first few mesh intervals (say 1,2,...,m) use one of the "standard" types of formulas, e.g., (29), (30), (31), but with a reduced value of ε (say $\tilde{\varepsilon}$). To indicate this we rewrite the general mesh spacing formula (8) as:

$$(x_{i+1} - x_i) = \frac{\Lambda}{F(x_i, x_{i+1}, v_i, v_{i+1}, \tilde{\varepsilon})} \qquad (35)$$

The examples so far have used an $\varepsilon = 1$. If an $\tilde{\varepsilon} = 1/16$ is used in an F containing $\sqrt{v}$ then the maximum possible mesh size is double what it is with $\varepsilon = 1$. We have previously mentioned some numerical experiments for which small values of ε have been used for all intervals. These experiments were generally unsuccessful. However, it was hoped that by using a reduced ε in only a few intervals behind the wave the difficulties associated with small ε would be minimal. This modification allows mesh points behind the wave to follow the wave more closely and so is analogous to Case IV (moving left boundary). Recall (Figure IVb) that this only delayed integration "hang up" due to the crossover effect. Also needed is a technique by which mesh points can be removed from the right boundary.

2) To permit mesh points ahead of the wave to move toward and out of the right boundary, the usual mesh spacing formula for the last interval $[x_{N-1}, x_N]$ was replaced by:

$$(x_2 - x_1) + (x_3 - x_2) + \ldots + (x_{m+1} - x_m) + (x_N - x_{N-1}) =$$

$$\Lambda \left\{ \frac{1}{F(x_1,x_2,v_1,v_2,\varepsilon)} + \frac{1}{F(x_2,x_3,v_2,v_3,\varepsilon)} + \ldots \right.$$

$$\left. + \frac{1}{F(x_m,x_{m+1},v_m,v_{m+1},\varepsilon)} + \frac{1}{F(x_{N-1},x_N,v_{N-1},v_N,\varepsilon)} \right\} \qquad (36)^*$$

Note that the standard ε (rather than the reduced $\tilde{\varepsilon}$) is used in (36). In our examples we have used $\varepsilon = 1$ and $\tilde{\varepsilon} = 1/16$. Consider the special case where $m = 1$ in (36). As v_1 and $v_2 \to 0$, $(x_2 - x_1) \to 2\Lambda$ (from (35)), and the contribution from the first term on the RHS of (36) approaches Λ. The last term on the RHS of (36) cannot exceed Λ. Thus, in order that (36) be satisfied $(x_N - x_{N-1})$ must go to zero (or even negative values if the last term on the RHS is less than Λ). A numerical test of this case was run. $(x_N - x_{N-1})$ approached zero asymptotically. Because $(x_N - x_{N-1})$ appears in the denominator of certain terms of the differential algebraic system (13), the integration hung up because of this ill conditioning.

If $m>1$ then $(x_N - x_{N-1})$ must eventually become negative as $v_1, v_2, \ldots, v_{m+1}$ become small. There may be some question as to the validity of the solution of system (13) (modified by (35) and (36)) for negative values of $(x_N - x_{N-1})$. This did not seem to be a problem in the examples attempted with this formulation. However, the integration for all these examples hung up before the wave approached the right boundary. After presentation of numerical results for a typical case of this modified formulation, we will discuss a correct method of treating vanishing mesh intervals at the right boundary.

*Because (36) directly couples the dependent variable at many mesh points (instead of just two) it is necessary to place this relation in the last row of (13). The last row of the Jacobian of (13) will consequently have many no zero elements. However, as previously noted special variants of the GEARIB routines have been developed to accommodate these cases.

Case VI

System (13) modified by (35) and (36)

$$N = 31$$

$$x_1 = 0$$

$$x_N = 2.0$$

$$\varepsilon = 1.0$$

$$\tilde{\varepsilon} = 1/16$$

$$m = 5$$

$$(x_{i+1} - x_i) = \frac{\Lambda}{\sqrt[4]{\frac{(v_i + v_{i+1})^2}{2} + \left(\frac{v_{i+1} - v_i}{x_{i+1} - x_i}\right)^2 \left[\frac{(v_i + v_{i+1})^2}{2(v_i^2 + v_{i+1}^2 + 10^{-30})}\right]} + \varepsilon} \qquad (37)$$

Of course (37) is modified in accordance with (35) for mesh intervals 1 to 5 and (36) is used for the last mesh interval. The numerical results are shown in Figures VIa, b, and c. Note the plotter malfunctions which caused the distortion at the beginning of Figure VIa and the skip near the end of Figures VIb and VIc.

From Figure VIb it can be seen that the trajectories for x_{N-1} and x_{N-2} have crossed x_N (= 2.0) before the integration hung up. Although there is a potential danger of the integration hanging up (due to a singularity in the governing system) when $x_{N-1} = x_N$, this did not occur for this case because the integration step size used at this point was such that $|x_N - x_{N-1}|$ did not become excessively small. Comparing Figure VIb with Figure Vb, we see that at least initially (up to about t = 0.40) our new formulation has the desired effect. Near t = 0.40, $(x_2 - x_1)$ is close to its maximum permissible value (= 2Λ) but then a slight spurious build up of v_2 (from 0.014 to 0.056) causes a dip in the x_2 trajectory. (This small change in v_2 has a noticeable effect on $(x_2 - x_1)$ because of the small value of ε (= 1/16) being used in (37).) The integration proceeds with both $(x_2 - x_1)$ and $(x_3 - x_2)$ near their maximum permissible values at about t = 0.61. In order for the mesh to continue to follow the wave, it is now necessary for $(x_4 - x_3)$ to

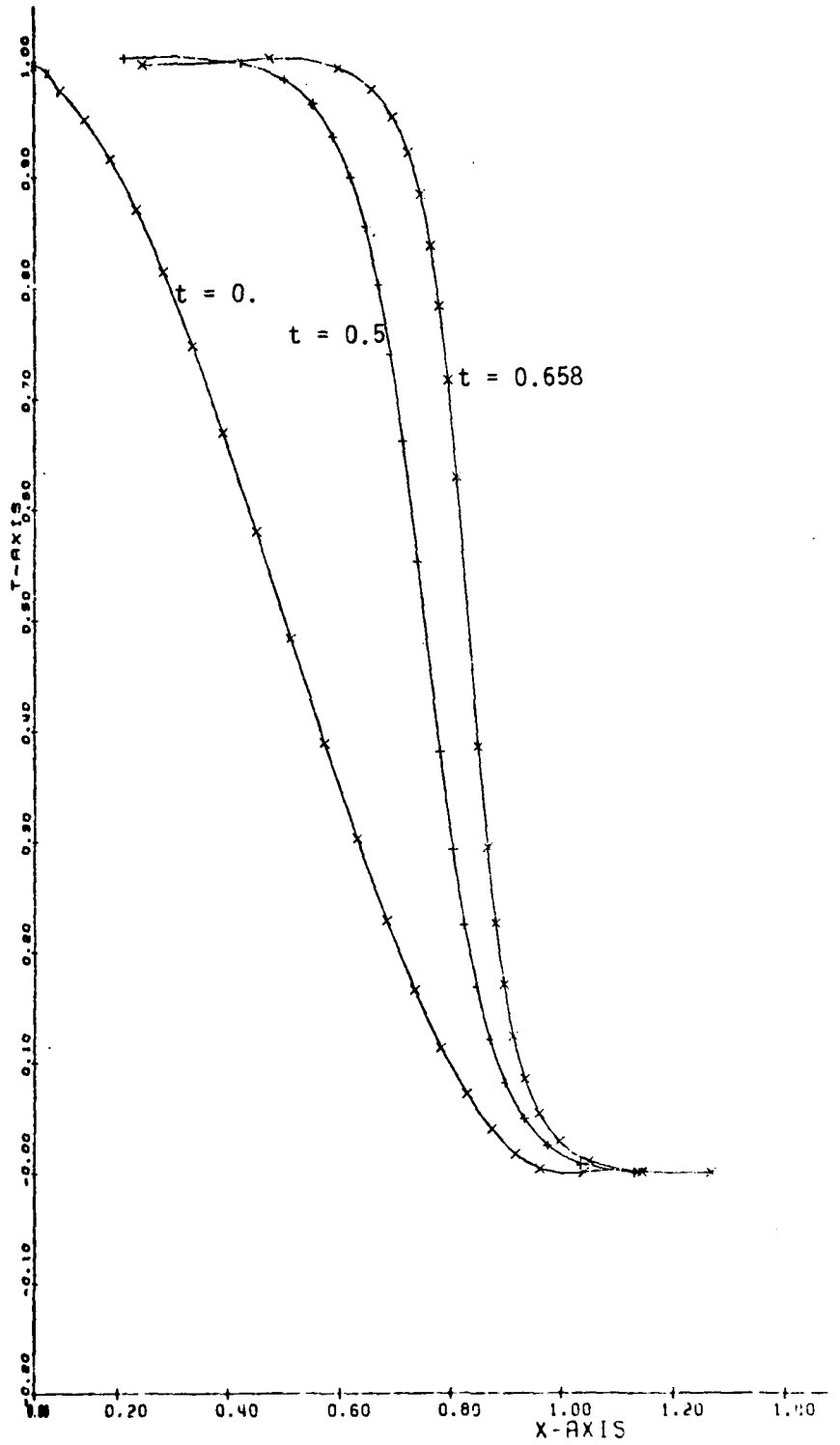


Figure VIa - T-Profiles for Case VI

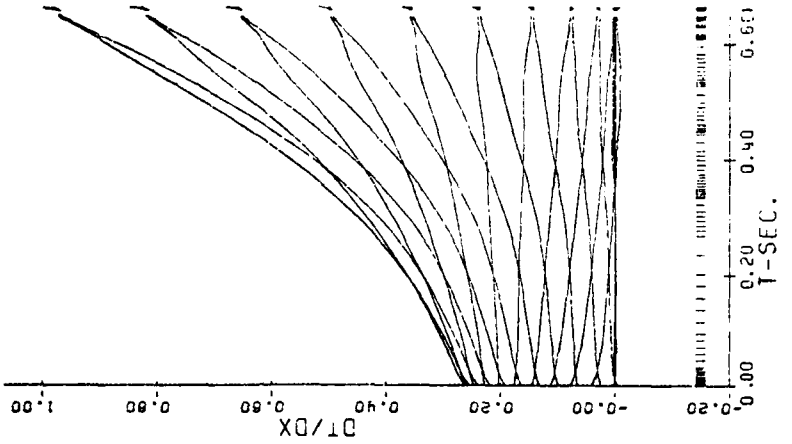


Figure VIc - v-Trajectories and Integration Step Sizes for Case VI

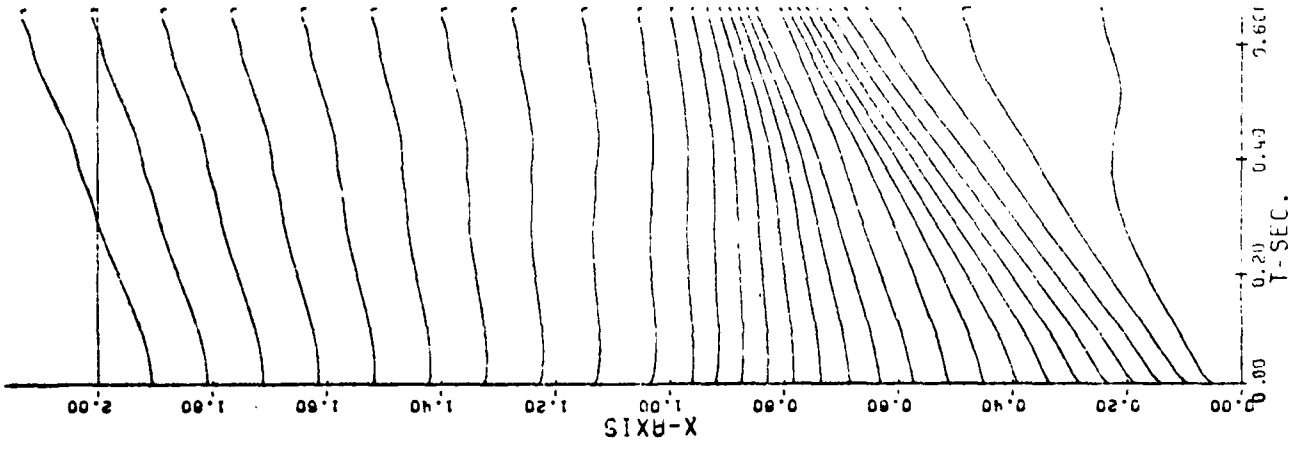


Figure VIb - x-Trajectories for Case VI

increase. However, v_3 has not fallen off enough to allow this. The only way to obtain the required mesh concentration in the crest is for a crossover to occur.

From Figure VIb, it may not seem that a crossover should be occurring since the mesh size in the middle of the wave is nowhere near its maximum permissible size. However, recall that the mesh spacing formula (37) contains the additional term $(v_i + v_{i+1})^2$. There is no way this term can become zero in the middle of the wave. What has happened is that v_i has become equal to v_{i+1} (see Figure VIc) reducing the second term in $\sqrt[4]{\ }$ of (37) to zero. This is the same effect that we have already seen in Cases II, III, and IV. Although the modified formulation has succeeded in delaying the first major crossover, when this crossover is required it is so violent that the integration cannot continue.

To eliminate the possibility that our questionable treatment of the problem (i.e., allowing the mesh size to become negative for the last interval) influenced the preceding conclusion, the same problem was solved in a different manner. Instead of allowing x_{N-1} to cross x_N, the integration continued until $x_{N-1} = x_N$* (to within the user specified integration accuracy). Then T_N, v_N, and x_N were eliminated from the system, and the integration was restarted with the boundary conditions now on T_{N-1} and x_{N-1} and with a formula similar to (37) now specifying the mesh spacing for the interval $(x_{N-1} - x_{N-2})$. The integration then continued until $x_{N-2} = x_{N-1}$ where another restart was made, etc. Until $x_{N-1} = x_N$, the numerical results were identical to those of Case VI. Upon restart, there was a slight change in slopes of the x_i and v_i trajectories, especially for the larger values of i. At t = 0.577, $x_{N-2} = x_{N-1}$, whereas in Case VI this did not occur until t = 0.631. The integration hung up at t = 0.659 versus t = 0.658 in Case VI. The T-profiles and the x_i and v_i trajectories (except for i close to N) were practically indistinguishable for the two cases.

*Near such singular points the integrator has considerable difficulty. The technique by which such singular points are determined could be improved by making use of the polynomial representation (in t) of each of the dependent variables which is generated by GEARIB. However, this was not our objective in this example so no effort was made in this direction.

Several other variations of this formulation were attempted. Instead of mesh spacing formula (37), formula (34) was used. This did not make any significant difference. Other cases were run using $\tilde{\varepsilon} = 1/256$ (instead of 1/16) in conjunction with both formulas (34) and (37). The integration managed to proceed somewhat farther (e.g., $t = 0.80$) but $(x_2 - x_1)$ was more erratic, as expected, because of the small value of $\tilde{\varepsilon}$.

E.7 Case VII

Of all the mesh spacing formulas discussed so far that of Case I (equation (27)) produced the best results. However, as discussed previously, this formula is clearly inadequate near the knees of the T-profile. It is almost certain that we need a term containing $(v_{i+1} - v_i)^2/(x_{i+1} - x_i)^2$, but so far all of our attempts to use this term (except for the "artificial" Case V) have been thwarted by the "crossover effect." This suggests combining the term $(v_i + v_{i+1})^2$ of formula (27) with $(v_{i+1} - v_i)^2/(x_{i+1} - x_i)^2$ to yield a mesh spacing formula with the desirable properties of Case I (i.e., allowing mesh points to move smoothly through the wave) with those of Cases II - VI (i.e., concentration of mesh points at the knees of the wave).

This has already been attempted in formula (37) of Case VI. One clear effect of the additional term $(v_i + v_{i+1})^2$ in (37) is to force a more concentrated mesh in the middle of the wave. (Compare Figure VIb with Figures IIb or IVb.) However, this increased mesh concentration has not allowed mesh points to pass smoothly through the wave. The integration in Case VI was still forced to stop because of the violent crossover effect associated with $(v_{i+1} - v_i)^2/(x_{i+1} - x_i)^2$. An examination of the relative contributions of the terms (37) showed that as the wave steepened the $(v_{i+1} - v_i)^2/(x_{i+1} - x_i)^2$ term generally dominated the $(v_i + v_{i+1})^2$ term, even in the middle of the wave where $(v_i + v_{i+1})^2$ is at its largest. The consequence is that the "easiest" way to provide the required increased mesh concentration at the crest and trough is to maximize the mesh interval at the center of the wave by making $v_i = v_{i+1}$, which is what happens at the end of the integration in Case VI (see Figure VIc) and also in Cases II, III, and IV.

It seems, then, that to prevent the crossover effect the contribution of $(v_i + v_{i+1})^2$ should be such that this term is dominant in the middle of the wave. The formula should also be such that $(v_{i+1} - v_i)^2/(x_{i+1} - x_i)^2$ is important at the knees of the wave. This suggests the following mesh spacing formula:

$$(x_{i+1} - x_i) = \frac{\Lambda}{\sqrt[4]{\left(\frac{v_i + v_{i+1}}{2}\right)^2 + \left(\frac{v_{i+1} - v_i}{x_{i+1} - x_i}\right)^2 \left[\frac{C\beta^2 \left(\frac{v_{i+1} - v_i}{x_{i+1} - x_i}\right)^2}{\gamma^2 \left(\frac{v_i + v_{i+1}}{2}\right)^2 + \varepsilon_1}\right] + \varepsilon}} \qquad (38)$$

Here γ and β are the coefficients in Burger's equation (3a), ε_1 is a constant to be set so that [ ] does not come too close to becoming singular, and C is a numerical "tuning" constant. If C = 1 and $\varepsilon_1 = 0$, the [ ] term approximates the square of the ratio of the diffusive term to the nonlinear term of Burger's equation. (Strictly speaking there should be a factor $(T_i + T_{i+1})^2/4$ in the denominator of [ ].) The rationale for (38) is that if the diffusive term is important (e.g., at the knees of the wave) then $(v_{i+1} - v_i)^2/(x_{i+1} - x_i)^2$ should have a significant influence in the mesh spacing formula. Accordingly, [ ] has been defined so that [ ] $\geq O(1)$ when the diffusive term is important.

Case VII

$$N = 31$$

$$x_1 = 0.$$

$$x_N = 2.0$$

$$\varepsilon = 0.5$$

$$\varepsilon_1 = 0.1$$

$$C = 0.25$$

$(x_{i+1} - x_i)$: formula (38)

The numerical results are shown in Figures VIIa, b, and c. Because of plot routine storage limitations, the plots of the x and v trajectories were terminated a few steps before the final station, t = 3.1. From the T-profiles it can be seen that

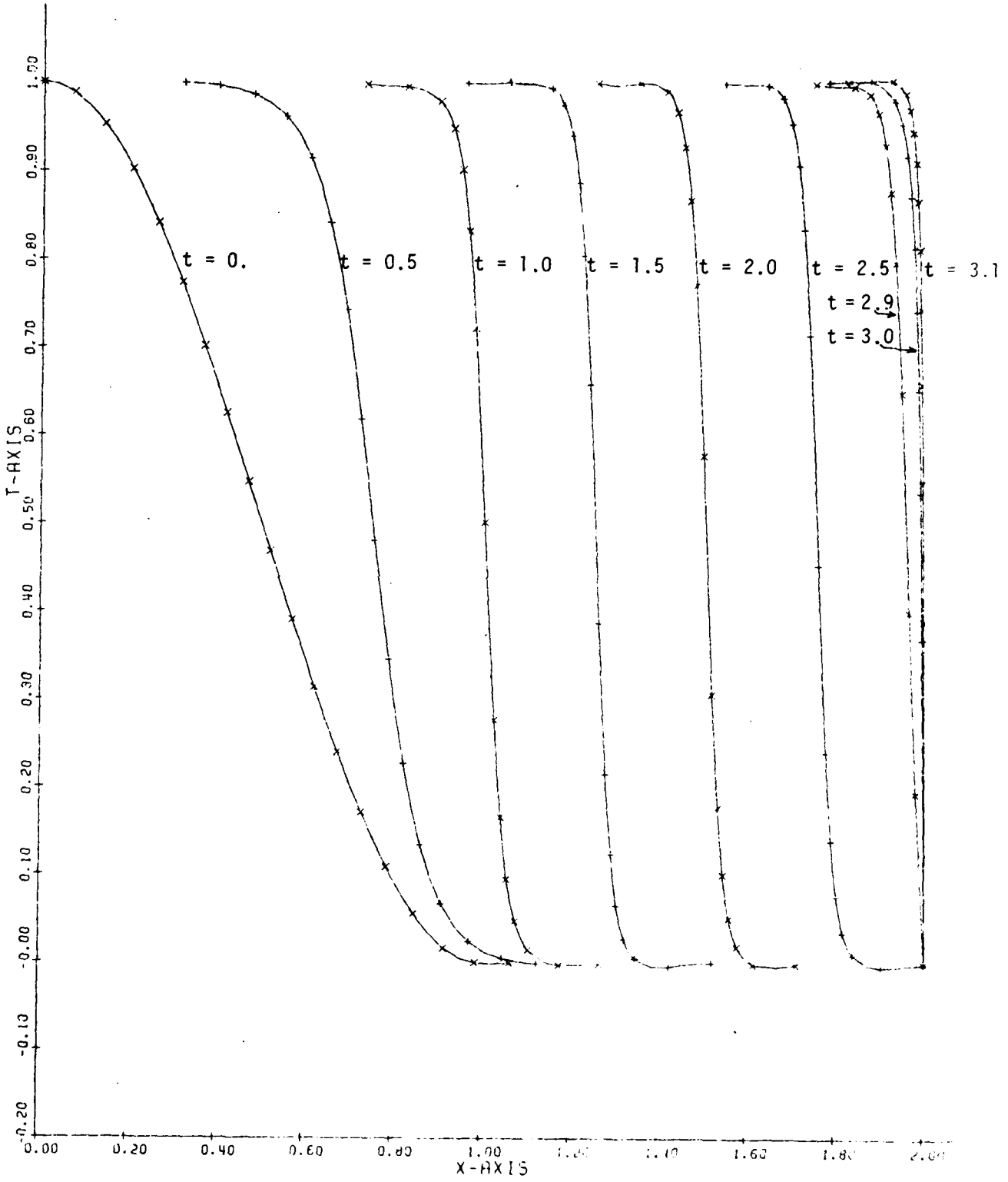


Figure VIIa - T-Profiles for Case VII

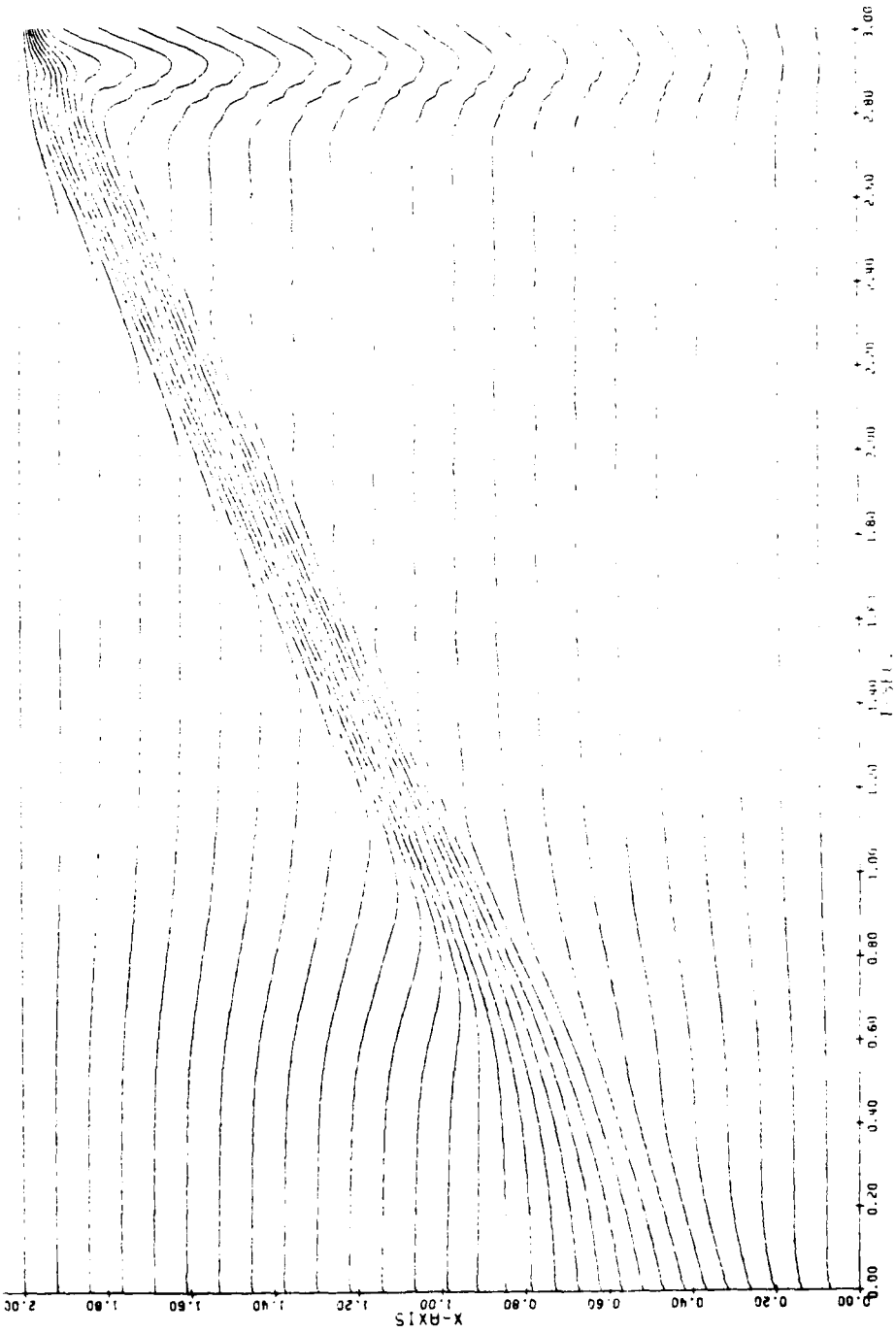


Figure VIIb - x-Trajectories for Case VII

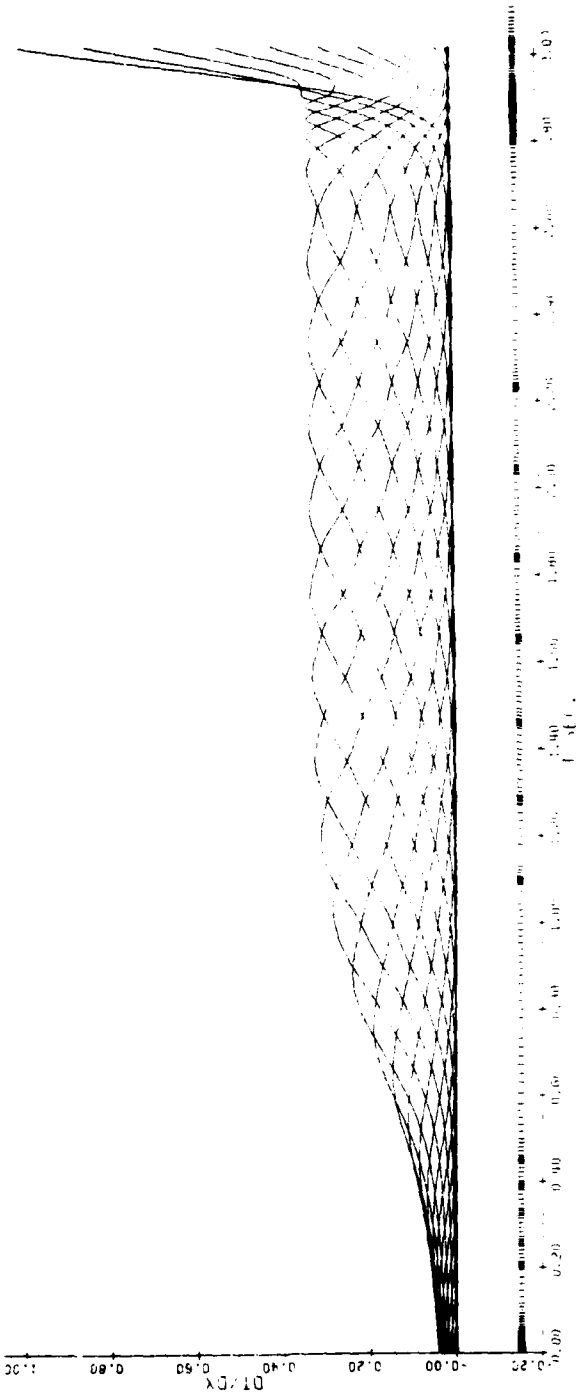


Figure VIIc - v-Trajectories and Integration Step Sizes for Case VII

t = 3.1 is nearly steady state. A later example (Case IX) will show the actual attainment of steady state.

A comparison of Figures VIIa and b with Figures Ia and b shows that formula (38) does in fact have the desired effect. The "crossover effect" has been eliminated, i.e., the mesh points are now able to pass smoothly through the wave. There is also an increased concentration of mesh points at the knees of the wave which eliminates most of the spurious oscillations in the T-profiles that are present in Figure Ia. The v-trajectories of Figure VIIc look much different than those of Figure Ic. This is due to the fact the integration proceeded slightly further in Case VII (t = 3.1 versus t = 2.7). The T-profiles steepened significantly between t = 2.7 and t = 3.1. Consequently, the scaling factor for the v-trajectories was quite different in the two cases.

A check of the relative importance of the terms in (38) shows that when the wave is traveling undistorted (from t = 1.0 to t = 2.5) the $(v_i + v_{i+1})^2/4$ term dominates the $[\ ]\ (v_{i+1} - v_i)^2/(x_{i+1} - x_i)^2$ term in the mesh interval at the "center" of the wave by 3:1. However, one or two mesh intervals to either side of this center interval the dominance is reversed.

Comparing the integration statistics for Case I and Case VII at t = 2.7, we find that Case VII is approximately 15% less efficient in terms of the number of corrector iterations required. Case VII is slightly more efficient in terms of the number of Jacobian evaluations required.

Several variations of Case VII were run. In one set of experiments, the effect of ε_1 was examined. Values of 1.0, 0.1, and 0.01 were used. The integration statistics for $\varepsilon_1 = 1.0$ were significantly worse than for $\varepsilon_1 = 0.1$. In the case of $\varepsilon_1 = 0.01$, the integration hung up early in the run. Based on these results $\varepsilon_1 = 0.1$ was used in all subsequent runs.

Another set of experiments tested the effect of varying ε. $\varepsilon = 0.5$ was found to be significantly better than $\varepsilon = 1$. An attempt to use $\varepsilon = 0.25$ failed to converge on the initial conditions. In another test, an ε that decayed in time was used ($\varepsilon = 0.5e^{-t}$). Initially, this was identical to Case VII. At t = 0.5 the variable ε case was statistically slightly better than Case VII. However, by the end of the run, its statistics were much worse than those of Case VII. The small value of ε at the end of the run did have the effect of forcing more mesh points

into the wave. For example, the T-profile at t = 3.0 of Figure VIIa shows 11 mesh points in the region $x \geq 1.8$, whereas the small ε case has 17 in the same region.

Another set of experiments examined the effect of using smaller values of β. Reducing β makes the T-profiles steeper and generally makes the problem more difficult. A repeat of Case VII with $\beta = 0.001$ (instead of 0.01) integrated very inefficiently and the T-profiles had significant oscillations. This case was rerun with the constant C in (38) increased by 100 to compensate for the smaller value of β. (The product $C\beta^2$ is now the same as for Case VII.) Up to about t = 0.1, this case behaved similar to Case VII. Beyond this the integration statistics degraded. The integration was stopped (deliberately) at t = 0.6. By this time, the T-profile was significantly steeper than for $\beta = 0.01$. (Maximum slope about three times as great.) The T-profile also had noticeable oscillations at t = 0.6

In an attempt to improve the numerical results for the $\beta = 0.001$ case, this case was rerun using a variant of the mesh spacing formula (38) in which the exponent of the $(v_i + v_{i+1})$ term in the denominator of [ ] was 4 (instead of 2). This modification reduced the spurious oscillations in the T-profiles. However, the integration statistics at t = 0.6 (the end of the run) were about the same as they were for an exponent of 2. This modified formula was also used for a $\beta = 0.0001$ case. The integration hung up at t = 0.51. Prior to t = 0.51, the T-profiles and x and v-trajectories were quite erratic. A rerun of this $\beta = 0.0001$ case using $(v_i + v_{i+1})^6$ in [ ] of (38) produced even worse results.

E.8 Case VIII

In this section we describe the numerical results for an example problem for which an exact analytical solution is known. It is taken from Sincovec and Madsen [8], who also solved this problem numerically. This exact solution can be written:

$$T_{ex}(t,x) = \frac{(0.1e^{-A} + 0.5e^{-B} + e^{-C})}{(e^{-A} + e^{-B} + e^{-C})} \qquad (39)$$

where

$$A \equiv -(x - 0.5 + 4.95t)/20\beta$$

$$B \equiv -(x - 0.5 + 0.75t)/4\beta$$

$$C \equiv -(x - 0.375)/2\beta$$

β is the coefficient of the diffusion term in Burger's equation (1). The coefficient, γ, of the nonlinear term is assumed to be unity in (39).

For our examples, the x-interval is taken to be [0,1] and the boundary conditions are:

$$v_1 = 0, \qquad T_N = T_{ex}(t,1) \tag{40}$$

The initial condition on T is (39) evaluated at t = 0. The initial conditions for x and v are computed in terms of this analytical representation of $T_{ex}(0,x)$ just as they were previously for the initial condition (24).

Strictly speaking the boundary condition $v_1 = 0$ is not consistent with (39). However, for the values of β being considered, $[\partial T_{ex}/\partial x](t,0)$ from (39) is less than 10^{-4}. Initially, $T_{ex}(t,1) = 0.1$ to eight figures. However, as the wave moves to the right this is no longer true. Consequently, the exact time dependent boundary condition (40) was used.

Case VIII

$$\begin{aligned} N &= 31 \\ x_1 &= 0. \\ x_N &= 1.0 \\ \beta &= 0.005 \\ \varepsilon &= 1.0 \\ \varepsilon_1 &= 1.0 \\ C &= 0.025 \end{aligned}$$

$$(x_{i+1} - x_i) = \Lambda \left\{ 0.05 + \frac{1}{\sqrt[4]{\left(\frac{v_i + v_{i+1}}{2}\right)^2 + \left(\frac{v_{i+1} - v_i}{x_{i+1} - x_i}\right)^2 \left[\frac{C\beta^2 \left(\frac{v_{i+1} - v_i}{x_{i+1} - x_i}\right)^2}{\gamma^2 \left(\frac{v_i + v_{i+1}}{2}\right)^2 + \varepsilon_1}\right] + \varepsilon}} \right\} \tag{41}$$

The numerical results are shown in Figures VIIIa, b, and c. Because of a flaw in the logic that determines the scaling factor for the v-trajectory plots, these trajectories have not been normalized so that the maximum value is unity. Also, because of memory limitations for the storage of plotter data, only every fourth integration point has been plotted in Figures VIIIb and VIIIc.

Note that the initial T profile in Figure VIIIa has regions of steep slope and high curvature, in contrast to the gentler initial T-profile used in previous examples. Decreasing β accentuates the slope and curvature of this initial profile. We had intended to use $\beta = 0.003$ in Case VIII in order to make a direct comparison of our numerical results with those of [8] (for which $\beta = 0.003$ was used). It turned out, however, that it was difficult to get the iterations for the initial x_i and v_i to converge for such small values of β. By adjusting the constants ε_1 and C in the mesh spacing definition and by adding the "lower bound" term, (0.05Λ), in (41) and by making some other adjustments in the initial condition iteration scheme, convergence could usually be attained. However, for $\beta = 0.003$, the initial distribution of mesh points was erratic and the initial v_i contained some rather large, $O(0.3)$, spurious oscillations. This was true in spite of the fact that the residuals (of all the algebraic equations that must be satisfied by the initial values of T_i, v_i, x_i and Λ) were driven to very small values. The integration was usually able to get started in such cases but always hung up for small t as the v_i oscillations built up, instead of dying off as they normally do. In the many attempts that were made to run this exact solution example for small values of β, $\beta = 0.005$ was the smallest value for which reasonable initial conditions and integration efficiency could be obtained.

The T-profiles of Figure VIIIa are smooth with no significant spurious oscillations and an apparently good distribution of mesh points. The final T-profile (at $t = 1.5$) was the constant steady state value, $T=1$, and was not plotted. The agreement between these profiles and the exact solution was good. Some of the largest discrepancies occurred for the profile at $t=1$:

$T_{Numerical}$	T_{Exact}
0.6072	0.5966
0.3663	0.3475
0.2574	0.2448

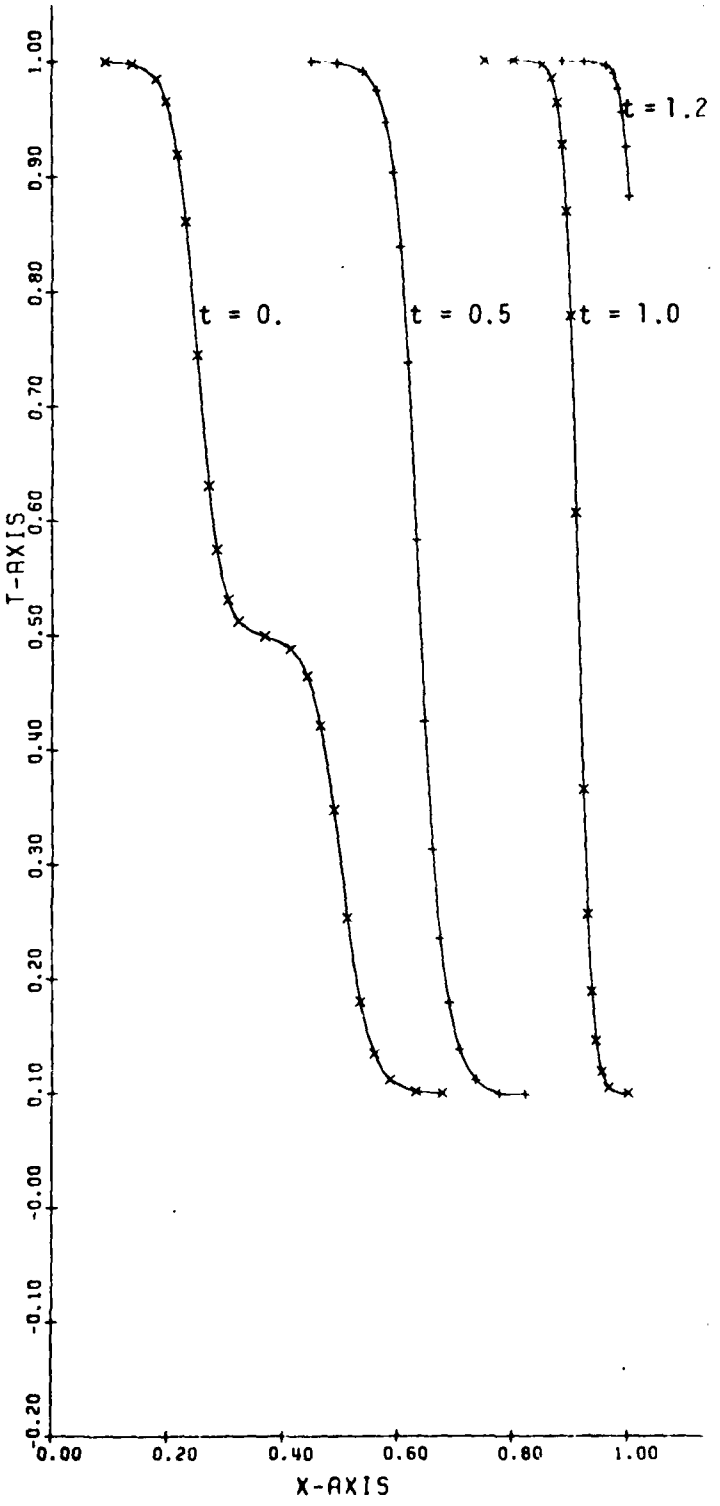


Figure VIIIa - T-Profiles for Case VIII

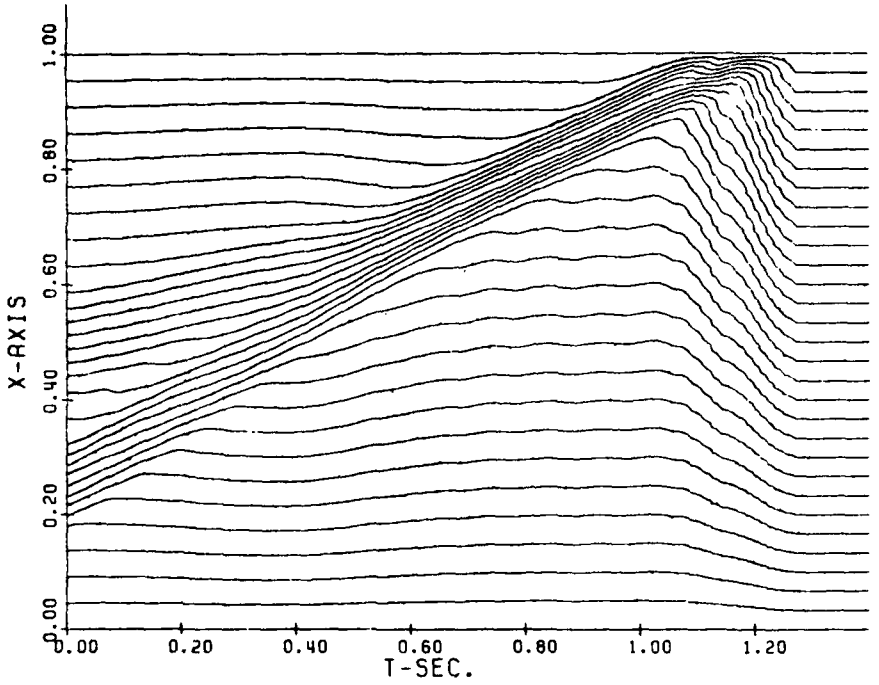


Figure VIIIb - x-Trajectories for Case VIII

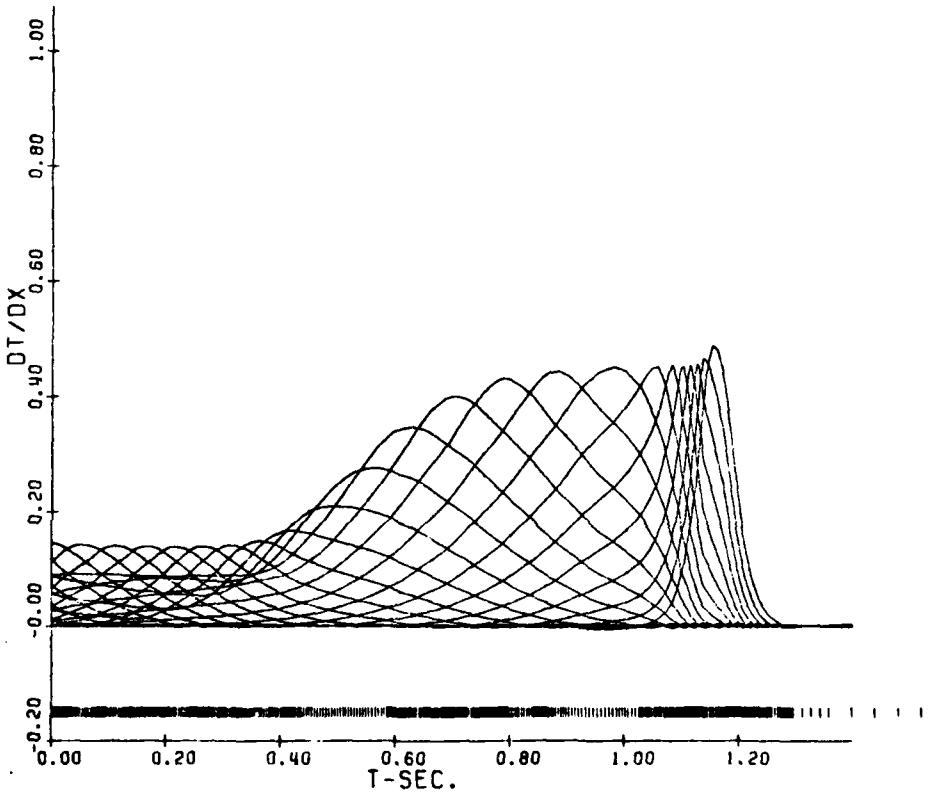


Figure VIIIc - v-Trajectories and Integration Step Sizes for Case VIII

For these values of T the slope is large (O(15)) and the mesh spacing is small (Δx = O(0.01)). Consequently, a plot of the exact solution would hardly be distinguishable from the profiles of Figure VIIIa.

The plots of the x and v-trajectories seem to be well behaved except possibly for the x-trajectories for small t and for 1.0 < t < 1.2. Shortly after t = 1.2 the steady solution has been reached.

From the plot of step sizes of Figure VIIIc, it is clear that the integration did not proceed efficiently. Not only were many steps taken but the ratio of corrector iterations to steps was high (NRE/NSTEP≈4) and the ratio of Jacobian evaluations to steps was high (NJE/NSTEP≈0.5). It is likely that these statistics could be improved somewhat by a better selection of the tuning parameters, ε, ε_1, and C. In Case VIII, these parameters were chosen to provide good convergence properties of the iteration for initial conditions. By smoothly altering them during the course of the integration (toward the values used in Case VII, say) a more efficient integration may have resulted.

The numerical results of reference [8] were generated by discretizing the standard Burger's equation (1) on a uniform mesh (200 points) using three point differencing. The time integration was performed by GEARB , a subroutine package similar to GEARIB. The numerical examples of [8] were for β = 0.003 and an integration error tolerance of 10^{-5}. (We used 10^{-4}.) These two differences make the numerical problem of [8] somewhat more demanding with respect to computation time.

The time integration in [8] was carried out to t = 1.1. A comparison of integration statistics with Case VIII at this time is:

	NSTEP	NRE	NJE
[8]	434	648	6
Case VIII	532	1909	268

The maximum error in T for the two cases was about the same (0.02). The minimum mesh size in Case VIII was about 0.007; slightly larger than the uniform 0.005 mesh size of [8]. (The 0.05 term in the mesh spacing formula (41) may have

influenced this minimum mesh size even though the minimum allowable mesh size was $0.05\ \Lambda \approx 0.0023$.)

Now [8] uses many more mesh points (200) than Case VIII which uses only 31. However, Case VIII has three dependent variables per mesh point, whereas [8] only has one. Thus, the ratio of the system sizes is about 2:1. A rough estimate then is that it takes [8] about twice as long to perform a single corrector iteration. On this basis alone [8] seems to have a definite advantage in terms of integration efficiency. Add to this the great disparity in the number of Jacobian evaluations and the fact that the problem solved in [8] was more demanding computationally. The conclusion is that the method employed in [8] has a clear cut advantage.

E.9 Case IX

In this section the results for a case similar to Case VIII are presented. Everything is the same except that the right boundary condition is held fixed, $T_N = 0.1$. The results are shown in Figures IXa, b, and c. Until about t=1, the results for the two cases are practically identical. At steady state the results are completely different as seen by a comparison of Figures VIII and IX. From about t = 1.1 to t = 1.3, the integration of Case IX proceeded significantly more efficiently. In Figure IXa the T-profiles for t = 1.2 and t = 1.5 are practically indistinguishable since t = 1.2 is nearly steady state. (The +'s correspond to t = 1.2 and the X's correspond to t = 1.5.) One interesting aspect of the final T-profile is that the minimum mesh spacing is O(0.003), considerably smaller than the minimum of O(0.007) of Case VIII. If Case IX were to be solved by the methods of [8], the advantage of [8] over the present method would not be so clear cut because of the increased number of mesh points required to obtain a mesh size of 0.003 on a uniform grid.

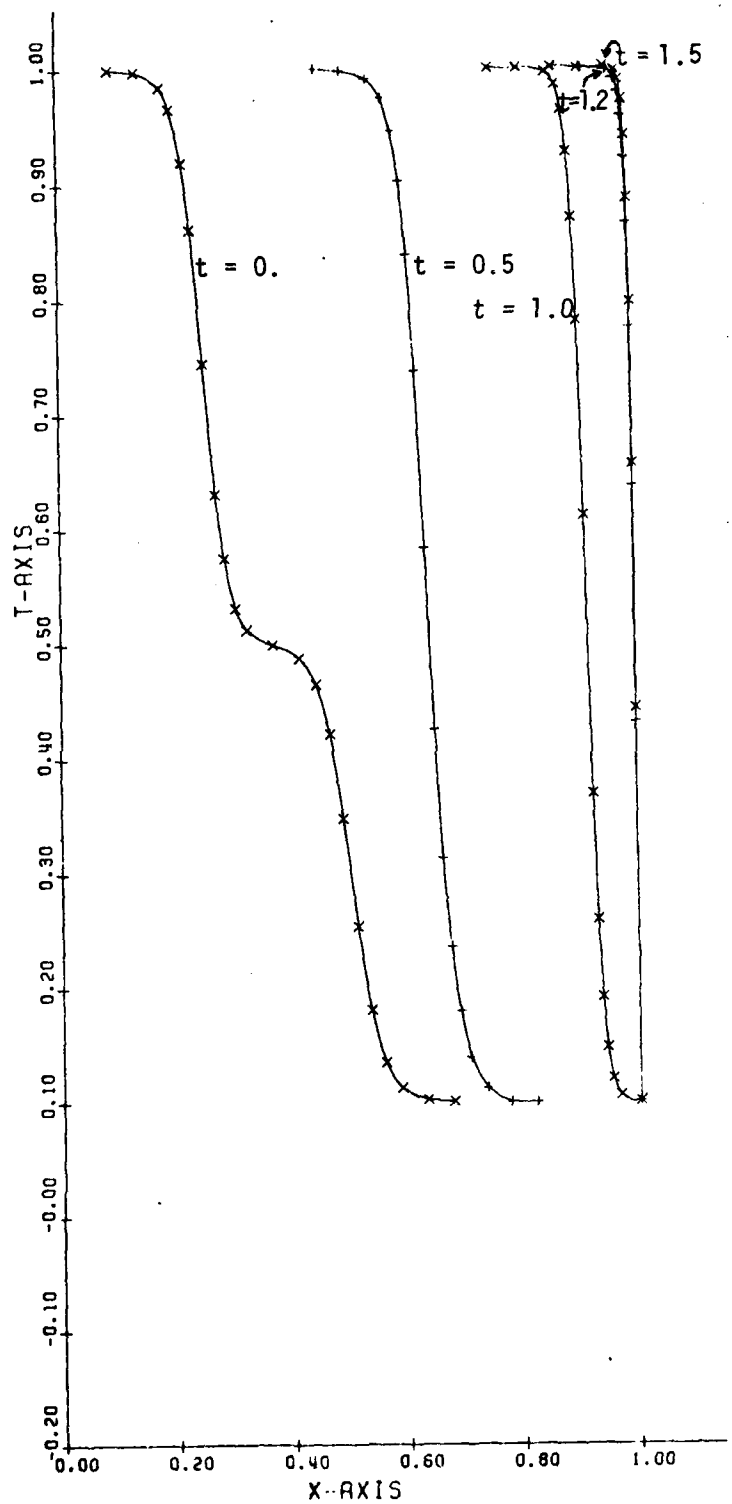


Figure IXa - T-Profiles for Case IX

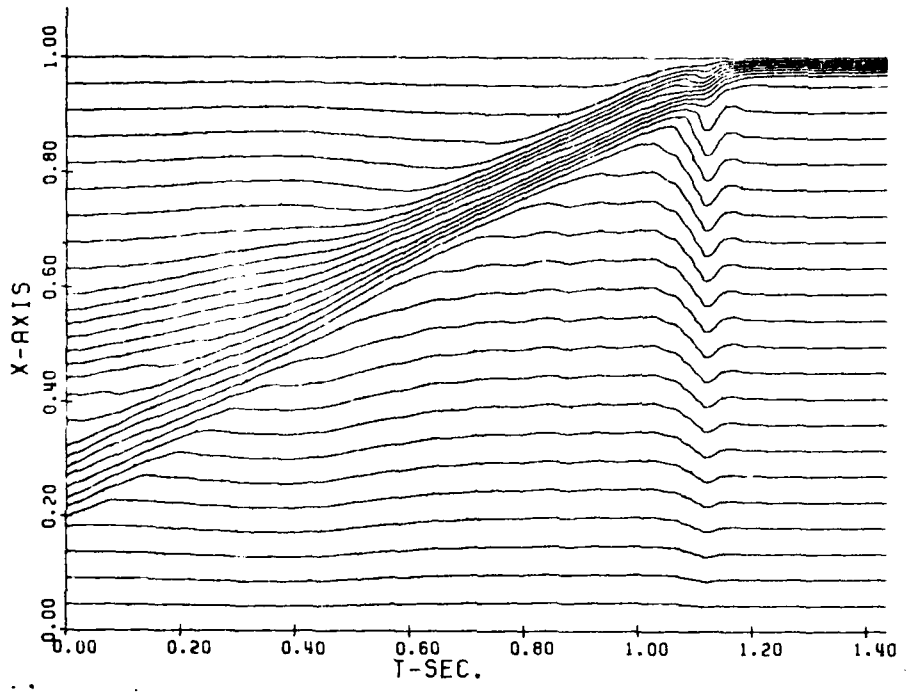


Figure IXb - x-Trajectories for Case IX

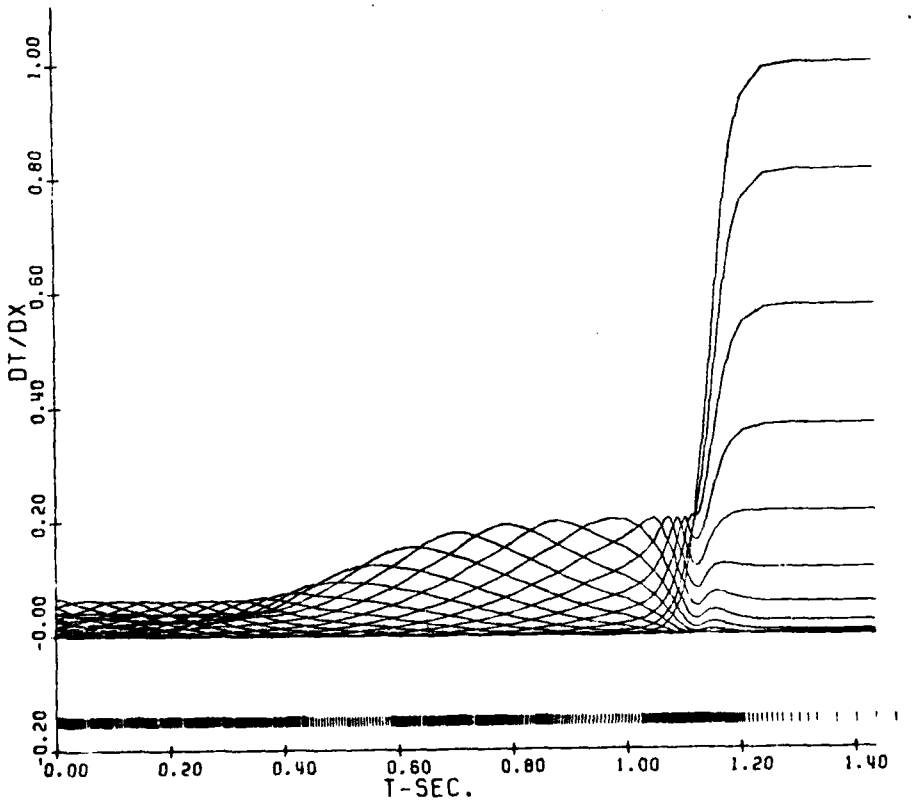


Figure IXc - v-Trajectories and Integration Step Sizes for Case IX

II. CONCLUSIONS AND RECOMMENDATIONS

A. Conclusions

During this study a variety of mesh spacing formulas have been used. The better ones, e.g., Cases VII, VIII, and IX did a reasonably good job of placing the mesh points where they were most needed. These formulas are based more on experiment and intuition than on theory. Despite this lack of theoretical foundation, the variable meshes so produced are far better than uniform meshes. For example, in Case VIII 31 mesh points produced about the same accuracy as 200 uniformly spaced points. Of course, for Burger's equation there are three dependent variables per mesh point for the formulation used in this study, versus one per mesh point for the usual formulation, e.g., reference [8]. For large partial differential systems, e.g., those involving many chemical dependent variables, the ratio (dependent variables/mesh points) would approach $2n$, where n is the number of basic dependent variables. Thus if a variable mesh scheme uses only half as many mesh points as a uniform mesh method for such problems, the former method would be competitive.

The foregoing conclusion is true providing the time integration can be performed as efficiently for variable meshes as it is for uniform meshes. However, this was not the case. In Case VIII the integration statistics were much worse than those for a similar case reported in [8]. One experiment performed during this study was to take a variable mesh case that was having integration difficulty (e.g., small and erratic step sizes, build-up of spurious oscillations, difficulty with corrector convergence), stop the integration at some point before "hang up," redefine the mesh spacing formula so that the mesh would stay fixed from then on, and then restart the integration. The results were dramatic; the oscillations died out and the integration proceeded smoothly. Of course, the fixed mesh soon became inadequate to describe the solution.

What is the reason for this integration difficulty? Most likely it is due to poor conditioning, especially when the T-profiles become steep and have regions of high curvature. This was already discussed in conjunction with Case VIII. Attempts were made to solve Case VIII with $\beta = 0.003$ (instead of $\beta = 0.005$ that was eventually used). Considerable difficulty in obtaining initial condition convergence was

encountered. Even when convergence was obtained, the x_i distribution was peculiar and $|v_i - v_{i\,exact}|$ * was relatively large. However, the residuals of the algebraic equations that were being solved had been driven to small values. This phenomenon is practically the definition of ill conditioning.

Another indication of this poor conditioning was the disappointingly large ratios (NRE/NSTEP)** and (NJE/NSTEP)** that occurred in most of our numerical examples. Such large ratios are symptomatic of ill-conditioning.

B. Recommendations

Although definite progress has been made in developing an automatic mesh varying method, certain obstacles (especially the conditioning problem) preclude writing a production program based on this method. Additional analysis and experimentation is required in the following areas:

1) Replace the banded matrix equation solver within GEARIB by one which also provides an estimate of the condition number of the matrix, e.g., subroutine SGBCO described in [9]. (This change may require extensive coding revisions of GEARIB to accommodate the different storage arrangement of SGBCO.) Study the effect of the mesh spacing formula (including any "tuning" constants) on the condition of the matrix used for the quasi-Newton corrector iterations.

2) Continue investigation of mesh spacing formulas. Consider introducing additional mesh point dependent variables for the purpose of estimating third and fourth derivatives of T (with respect to x) in order to compute a mesh spacing formula more consistent with the theoretical error estimates of equation (19).

*Here $v_{i\ exact}$ is the derivative with respect to x of equation (39) evaluated at x_i.

**NSTEP: Number of successful integration steps
NRE: Number of corrector iterations
NJE: Number of Jacobian evaluations

3) From the standpoint of integration efficiency our most successful example was Case V. What made this case run efficiently was that the mesh points moved with the wave rather than passing through it. Consequently, once the wave established itself, the dependent variables were either constant or linear in time. This suggests developing methods by which mesh points are added to (deleted from) the field as they are needed (unneeded). By so doing, shifting of the entire set of mesh points just to transfer a single mesh point from one place to another would be averted. (This was seen most dramatically in the "crossover effect" of Figures IIb, IIIb, and IVb, but it is also evident to a lesser extent in Figures Ib, VIIb, VIIIb, and IXb.) These mesh point additions (deletions) might be done at discrete times, in which case the method resembles the regridding approaches of references [5] and [6]. Preferably, this would be accomplished in a continuous manner by allowing a single mesh point to split in two or by two mesh points coalescing into one.

4) One of the assumptions throughout this study was that x_1 would coincide with the left boundary (say x_L) of the mathematical problem and x_N with the right boundary (say x_R), and that the number of mesh points N would be constant. According to formula (10), Λ then varies as the solution evolves. Now the spatial truncation error is related to Λ (equation (21)). A preferable approach would be to specify the desired level of truncation error (and therefore Λ) and then determine the number of mesh points N and/or the length of the finite difference mesh $[x_1, x_N]$ consistent with this specification. If $x_1 < x_L < x_2$ and/or $x_{N-1} < x_R < x_N$, the boundary conditions at x_L and x_R would be expressed in terms of values at two consecutive mesh points instead of the simpler expressions of system (13). This would increase the bandwidth at the beginning and end of the Jacobian matrix. However, special formulas could be developed which would allow solution of such modified matrices with very little additional cost. A simplification arising out of such a formulation is that Λ is not a variable, and so the modification of the GEARIB banded equation solver to accommodate a possibly full last column would no longer be required.

5) Ultimately, it is desired to extend these variable mesh methods to gas dynamic problems involving chemistry and radiation. Even "standard" gas dynamic problems exhibit phenomena not present in solutions to Burger's equation, e.g., waves traveling at variable speeds, waves being reflected from boundaries, waves traveling in opposite directions and impinging on each other. Thus, even after variable mesh methods are perfected for Burger's equation, there is still much work to be done before these methods can be successfully applied to realistic problems.

III. REFERENCES

1. Schimke, J. T., Rushmore, W. L., and Zelazny, S. W., "Numerical Model Development for Laser Cavity Flowfields," Interim Report for the period May 31, 1977 to May 31, 1978, for AFOSR Contract F49620-77-C-0076.

2. Hindmarsh, A. C., "GEARIB: Solution of Implicit Systems of Ordinary Differential Equations with Banded Jacobian" (Preliminary Documentation), UCID-30130, Lawrence Livermore Laboratory, February 1976.

3. Schimke, J. T., "Numerical Model Development for Laser Cavity Flowfields," Interim Report for the period May 31, 1978 to May 31, 1979, for AFOSR Contract F49620-77-C-0076.

4. Dwyer, H. A. and Sanders, B. R., "Modeling of Unsteady Combustion Phenomena," AIAA Paper 77-136, Los Angeles, California, 1977.

5. Chong, T. H., "A Variable Mesh Finite Difference Method for Solving a Class of Parabolic Differential Equations in One Space Variable," SIAM J. Num. Anal., Vol. 15, 1978, pp. 835-857.

6. Miller, J. V., Morton, K. W., and Baines, M. J., "A Finite Element Moving Boundary Computation with an Adaptive Mesh," J. Inst. Math. Applics., Vol. 22, 1978, pp. 467-478.

7. Cole, J. D., "On a Quasi-Linear Parabolic Equation Occurring in Aerodynamics," Quart. Appl. Math., Vol. 9, 1951, pp. 225-236.

8. Sincovec, R. F. and Madsen, N. K., "Software for Nonlinear Partial Differential Equations," ACM Trans. Math. Software, Vol. 1, 1975, pp. 232-260.

9. Dongarra, J. J., Moler, C. B., Bunch, J. R., Stewart, G. W., "LINPACK Users' Guide," SIAM, 1979.